

РОССИЙСКАЯ АКАДЕМИЯ НАУК
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

СООБЩЕНИЯ ПО ПРИКЛАДНОЙ МАТЕМАТИКЕ

М.В. КОЗЛОВ, Ю.Е. МАЛАШЕНКО,
И.А. НАЗАРОВА, А.Ф. РОНЖИН

**АНАЛИЗ РЕЖИМОВ УПРАВЛЕНИЯ
ВЫЧИСЛИТЕЛЬНЫМ КОМПЛЕКСОМ
В УСЛОВИЯХ НЕОПРЕДЕЛЁННОСТИ**

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР РАН
МОСКВА 2011

УДК 519.85

Ответственный редактор
член-корр. РАН, доктор физ.-матем. наук
Ю.А. Флеров

Рассмотрена проблема управления одностадийной системой исполняющих вычислительных модулей, при неизвестной длительности выполнения заданий, поступивших на обработку. Исследуются два режима управления выполнением заданий: последовательный и параллельный. Анализ работы системы производится с точки зрения пользователей. Случайное время обработки каждого задания и место заявки в очереди рассматриваются как неопределённость. В работе подробно описаны численные эксперименты, проведённые с помощью имитационного моделирования, приводятся графики и таблицы. Полученные результаты могут послужить математической (аналитической) основой для разработки диспетчерского программного обеспечения специализированных вычислительных комплексов.

Ключевые слова: распределённые вычисления, математическое моделирование, теория расписаний, порядковые статистики.

Публикуется в авторской редакции.

Рецензенты: В.В. Лаврентьев,
А.В. Лотов

Научное издание
© Учреждение Российской академии наук
Вычислительный центр им. А.А.Дородницына, 2011

Введение

В книге [1] В.Феллером формулируется следующая проблема: "Человек хочет открыть свою дверь. У него n ключей, из которых только один подходит к двери. По причине, о которой можно только догадываться, он пробует ключи случайно, так что при каждой попытке каждый ключ имеет вероятность быть выбранным $1/n$ и все возможные исходы, отвечающие данному числу попыток равновероятны". Впоследствии описанная проблема получила название "задача о ключе". Данную постановку в дальнейшем обобщали, видоизменяли и наполняли другим содержательным смыслом, например, человек мог иметь несколько связок ключей, дверь может быть открыта не одним ключом, в двери два и более замков, и т.д.

В современной практике применения высокопроизводительных вычислительных систем большую группу составляют потоковые задачи обработки информации [2], многие из которых можно рассматривать как варианты задачи о ключе. В отечественной практике для решения таких задач используются массово-параллельные системы [3]. Существуют успешные примеры реализаций программных комплексов, например, программный комплекс "Пирамида" [4, 5]. Обязательным элементом таких комплексов является пакетирование фрагментов, что связано как с сокращением процедур обмена, так и с обеспечением безотказности и упрощением управления в условиях гетерогенности аппаратно-программной среды [6-10].

В [11] рассматривается близкий по смыслу класс задач, алгоритмические процедуры решения которых можно определить как случайный перебор с неизвестным исходом. Суть проблемы состоит в следующем: из имеющегося набора данных требуется выявить фрагмент с уникальными свойствами. Для выполнения этой задачи необходимо реализовать одну и ту же процедуру (алгоритм) при различных начальных данных, раз-

битых в свою очередь на неделимые, содержательно значимые фрагменты. Если при выполнении задания, удаётся определить такой уникальный фрагмент, то говорят, что задача решена, и обработка задания прекращается. Если же при выполнении просматриваются все предъявленные данные и найти уникальный фрагмент не удаётся, то задание считается выполненным, но задача не имеет решения.

Процесс выявления уникального фрагмента проходит следующие основные этапы:

- создание и/или рекуррентное описание большого массива отдельных фрагментов данных;
- формулировка правила, по которому фрагмент объявляется уникальным,
- выбор наугад фрагмента из массива,
- проверка его уникальности.

При реализации вычислительных процедур поиска уникального фрагмента необходимо учитывать следующие особенности исходной проблемы:

- допускается разбиение массива данных на отдельные, даже пересекающиеся подмножества, таким образом, что решение исходной задачи будет получено в результате решения подзадач для подмассивов [11];
- для завершения задачи не всегда требуется обработка всех наборов данных, зачастую достаточно найти один уникальный фрагмент;
- содержательная постановка задач предопределяет проведение заданных вычислительных работ, и не предполагает календарного планирования при распределении вычислительного ресурса.

Задачи, вернее, наборы данных, поступают в произвольные моменты времени, имеют разный тип, но каждая из них требует большого, точно не известного, объёма вычислений. В об-

щем случае задана только верхняя оценка для величины затрат вычислительных ресурсов необходимых для выполнения конкретного задания; имеет ли решение соответствующая содержательная задача — неизвестно. Таким образом при решении задач возникает неопределённость, связанная как с составом заданий, так и с длительностью их обработки. Рассматриваемые задания будем далее обозначать: citu-задания, citu-работы, citu-задачи; от английского термина: *computationally intensive task (under) uncertainty*.

Citu-задания должны выполняться на специализированной вычислительной системе (СВС) в режиме реального времени. Для большинства из них задаются директивные сроки окончания (ДСО). Превышение ДСО означает, что решение соответствующей задачи утратило актуальность, вычислительные ресурсы — время, затраченное на её выполнение, — записываются в производственные потери. Кроме того, в системе возможно появление задач, имеющих подавляющий приоритет, т.е. требующих срочного решения вне зависимости от того, какие задачи в данный момент обрабатывает СВС. В этом случае решение всех имеющихся задач прерывается, и вновь поступившая выполняется от начала до конца в монопольном режиме СВС, после чего осуществляется завершение всех снятых со счёта заданий.

Все citu-задания одинаково важны и должны быть решены — каждая в отдельности и все в совокупности — как можно быстрее. Таким образом, citu-задания равнозначны, а пользователи, предоставляющие задачи для обработки, равноправны в рамках правил распределения вычислительных ресурсов. Поэтому стратегии управления не должны быть дискриминирующими по отношению к той или иной группе заявок. В данном случае под справедливым дележом ограниченного ресурса между равноправными пользователями будем понимать рас-

пределение, при котором ни один из партнеров не может улучшить свои показатели, не ухудшив соответствующие показатели для других своих коллег. Предполагается, что пользователи не являются антагонистами и стремятся к достижению некоторой общей корпоративной цели.

В работе анализируются две стратегии выполнения *citu*-заданий:

- последовательная, когда задания выполняются СВС в монопольном режиме, одно за другим;
- параллельная, при которой все задания выполняются одновременно.

В условиях уже упоминавшейся неопределенности исследуются индивидуальные временные показатели выполнения каждого задания при пакетной динамической обработке целого набора персонифицированных задач.

Настоящая работа является продолжением исследований, начатых в [12], и организована следующим образом. В §1 представлена модель абстрактной вычислительной системы и процесса выполнения однотипных *citu*-заданий. В §2-4 определяются стратегии выполнения предписанных работ и проводится анализ результатов имитационного моделирования. В §5, с позиции теории вероятностей, исследуются вопросы получения и динамического пересчёта прогнозных оценок возможностей решения *citu*-задач.

§1. Общее описание моделей

Для формализации описания процедуры выполнения каждого *citu*-задания воспользуемся теоретико-вероятностной моделью — случайной выборкой без возвратов [1]. В рамках стандартной модели определим каждое *citu*-задание как урну с чёрными и белыми шарами, а процесс его выполнения — как извлечение шаров из урны и определение их цвета. Под специализированной элементарной вычислительной операцией (СЭВО) будем понимать вытягивание из урны одного шара и идентификацию его цвета.

Если урна содержит только белые шары, то процесс прекращается после извлечения последнего шара; в реальности это соответствует сбою в СВС, или неправильной подготовке данных для *citu*-задания. В этом случае задание объявляется завершённым, а задача — не имеющей решения. Если в урне содержится хотя бы один чёрный шар, то после извлечения первого чёрного шара выполнение задания прекращается, и задача считается решённой.

Пусть указанные *citu*-работы и соответствующие им СЭВО выполняются на некоей гипотетической специализированной вычислительной системе (СВ-системе), которая состоит из центрального управляющего устройства (ЦУП-устройства) и набора единичных вычислительных модулей (ЕВ-модулей). ЦУП-устройство координирует работу всех ЕВ-модулей, организует и контролирует процесс выполнения заданий, поступающих от пользователей. Каждый ЕВ-модуль будем рассматривать как некий абстрактный безотказный вычислитель со своим блоком управления и памятью, но с неизвестной архитектурой и конструктивными особенностями. Производительность ЕВ-модуля определим как число выполняемых СЭВО в единицу времени.

Считается, что СВ-система может одновременно обрабаты-

вать как одно, так и несколько заданий на всех работоспособных в данный момент В-модулях. В свою очередь предполагается, что отдельное задание может быть разделено на подзадания, каждое из которых может выполняться независимо от других на отдельном ЕВ-модуле. При этом на каждом ЕВ-модуле в единицу времени может обрабатываться только одно задание или его часть.

Итак, под СВ-системой $M = \{M_0, P_1, \dots, P_i, \dots, P_m\}$ будем понимать набор ЕВ-модулей $P_i, i = \overline{1, m}$, каждый из которых выполняет определённое число v_i СЭВО в единицу времени, $i = \overline{1, m}$; и ЦУП-устройство — M_0 , распределяющее задания для обработки по имеющимся ЕВ-модулям и осуществляющее контроль за их выполнением. В рамках данной модели будем считать, что все ЕВ-модули являются идентичными и имеют равную мощность v .

Обозначим через V суммарную мощность СВ-системы, $V = \sum_{i=1}^m v_i = mv$. Далее будем говорить, что задание выполняется в монопольном режиме, если на его обработку направлены все ЕВ-модули, т.е. вся мощность системы.

Рассмотрим ситуацию, когда задания поступают от пользователей одновременно в виде некоторого набора, без каких-либо признаков или отношений предшествования между ними, т.е. пусть в момент времени $t = 0$ в вычислительную систему одновременно поступило множество J , состоящее из n заданий, т.е. $J = \{J_1, \dots, J_j, \dots, J_n\}$, где $j = \overline{1, n}, n \geq 2$. Считается, что каждое задание из набора может быть поставлено на обработку в любое время. Предполагается, что обработка J_j описывается случайной выборкой без возвращения, при этом урна содержит $\mathbf{w}_j, j = \overline{1, n}$ шаров, т.е. $\mathbf{w}_j$ — максимально возможное число СЭВО, необходимое для выполнения J_j .

Введем следующие обозначения:

$\mathbf{T}_j$ — нормативное время выполнения j -го задания в монополь-

ном режиме при условии, что необходимо осуществить полный перебор всех шаров урны, т.е. $T_j = w_j/V$;

X_j — абсолютная длительность — время, затраченное на решение j -й задачи в монопольном режиме. X_j — реализация случайной величины с неизвестной функцией распределения на $[0; T_j]$ для конкретного задания j и мощности V . Значение X_j становится известно только после завершения J_j , и, поскольку для его выполнения был задействован весь имеющийся вычислительный ресурс, то X_j — минимальное время решения j -й задачи в данной СВ-системе мощности V ;

X — множество значений X_j для фиксированного набора J' , т.е. $X = \{X_j\}, j = \overline{1, n}$;

Будем считать, что процесс обслуживания может начаться с выполнения как одного, так и нескольких заданий из имеющегося множества J , а перераспределение ресурса (ЕВ-модулей) производится только в момент завершения хотя бы одного задания. Выполненное задание покидает систему.

Обозначим через $\pi(S(J), E, t)$ — диспетчерское правило (стратегию), которое определяет порядок $S(J)$ выполнения заданий из множества J и число ЕВ-модулей e_j , выделенных для каждой j -й задачи в момент времени t . Здесь e_j — компонента вектора E , описывающего распределение всего ресурса по заданиям в момент времени t , $e_j \geq 0$, $E = (e_1, \dots, e_j, \dots, e_n)$. Таким образом $\pi(S(J), E, t)$ однозначно определяет размещение заданий по ЕВ-модулям в каждый момент времени t . Обозначим t_j^π — время пребывания j -й задачи в системе, т.е. длительность временного интервала от момента поступления задания в систему до момента, когда оно выполнено и покидает систему.

Сравнение различных стратегий $\pi(S(J), E, t)$ проведем на основании анализа временных характеристик выполнения заданий, описанных ниже. Для их формализации рассмотрим некоторый конкретный набор заданий J' . Предположим, что

для всех J_j из набора J' значения $\mathbf{X}_j$ и др. стали известны заранее, зафиксированы и остаются неизменными в серии последующих умозрительных экспериментов. Тогда для данного фиксированного набора J' можно вычислить:

— суммарное абсолютное время обработки всех заданий

$$U = \sum_{j=1}^n X_j;$$

— среднее абсолютное время выполнения одного задания

$$\bar{X} = \frac{1}{n} U = \frac{1}{n} \sum_{j=1}^n X_j.$$

Рассмотрим следующие отношения:

$$\frac{X_j}{\bar{X}} = \xi_j, \quad \frac{t_{jk}^\pi}{\mathbf{X}_j} = \eta_{jk}^\pi, \quad (1)$$

где t_{jk}^π — время пребывания j -го задания в СВ-системе согласно плану π , при условии, что оно покинуло её k -м по порядку.

Назовем безразмерную величину ξ_j показателем относительной длительности исполнения задания J_j для набора J' , η_{jk}^π — коэффициентом относительного времени запаздывания для J' при использовании стратегии π . Действительно, численное значение ξ_j показывает на сколько время обработки j -го задания отличается от средней длительности исполнения задания из J' , а η_{jk}^π — во сколько раз время пребывания в системе j -го задания при использовании стратегии π оказывается больше его абсолютного времени решения, т.е. минимального времени, за которое его можно завершить, используя всю имеющуюся мощность СВ-системы.

Одним из возможных и часто используемых критериев оценки стратегий является математическое ожидание (среднее значение) среднего времени пребывания заданий в системе

$\widehat{T}^\pi = \frac{1}{n}(t_1^\pi + \dots + t_n^\pi)$. Однако такая оценка является малоинформативной при анализе текущей ситуации и выполнении конкретного набора citu-работ.

Целью данной работы является исследование того, как стратегия, выбранная диспетчером для выполнения citu-заданий, т.е. распределение вычислительных ресурсов, влияет на значения η_j^π для конкретной заявки из фиксированного набора. Для каждого фиксированного набора J' следует изучать множество возможных значений $\widehat{T}^\pi = \langle t_1^\pi, \dots, t_n^\pi \rangle$, чтобы проанализировать как выполняются принципы равноправия пользователей и равнозначности их citu-заданий.

Далее рассмотрим две простейшие стратегии — последовательную и параллельную. Их описание помещено ниже, в §2, 3. Сравнение будем производить на одних и тех же наборах заданий, при этом будем полагать, что все задания имеют одинаковый объём $\mathbf{w}_j = \mathbf{w}$, и для всех J_j из набора J' численные значения $\mathbf{X}_j$ известны заранее, например, определены в ходе предварительных тестов. В частности $\mathbf{X}_j$ для каждого $j = \overline{1, n}$, является конкретной реализацией случайной величины, имеющей неизвестную функцию распределения $F_j(x)$ на отрезке $[0; \mathbf{T}]$. В §2, 3 предполагается, что каждая задача имеет решение, т.е. в каждой урне содержится хотя бы один чёрный шар.

§2. Последовательное выполнение заданий

Рассмотрим простейшую стратегию выполнения заданий из набора J , состоящую в следующем. Из набора заданий J выбирается случайным образом одно, допустим, j -е, и его обработка начинается в монопольном режиме на всех работоспособных ЕВ-модулях. После завершения J_j из оставшегося множества $J \setminus \{J_j\}$ случайным образом выбирается следующее задание и также обрабатывается всей системой. Случайный выбор задания для обработки можно мотивировать тем, что согласно предположению, высказанному в §1, все $J_j \in J$ равноправны, имеют равный объём, а реальное время обработки становится известно только после их завершения. Процесс продолжается до тех пор, пока все n задач множества J не будут решены. Другими словами, можно считать, что в начальный момент времени все задания из J по жребию образуют некоторую очередь, и в дальнейшем выполняются последовательно, одно за другим.

Описанную стратегию будем называть последовательной схемой обработки заданий и обозначать $seq(S(J), E, t)$. При этом число ЕВ-модулей, выделенных в определённый момент времени t для всех задач, кроме одной, пусть j -й, равно нулю, а $e_j = m$, т.е. $\forall l = \overline{1, n}, l \neq j, e_l = 0, e_j = m$, и $E = (0, \dots, 0, m, 0, \dots, 0)$. Такое распределение ресурса для j -го задания схематично изображено на рис.1.

Рассмотрим фиксированный набор заданий J' . Обозначим через $\hat{S}(J')$ множество всех возможных очередей, которые можно сформировать из заданий множества J' для обработки СВ-системой. Тогда $\hat{s}_i$ — отдельный элемент множества $\hat{S}(J')$ — вектор, каждая i -я из n компонент которого указывает, какое из n заданий набора J' должно выполняться i -м по порядку. Всего $\hat{S}(J')$ содержит $n!$ различных очередей-перестановок. Предположим, что к каждой из них была применена последо-

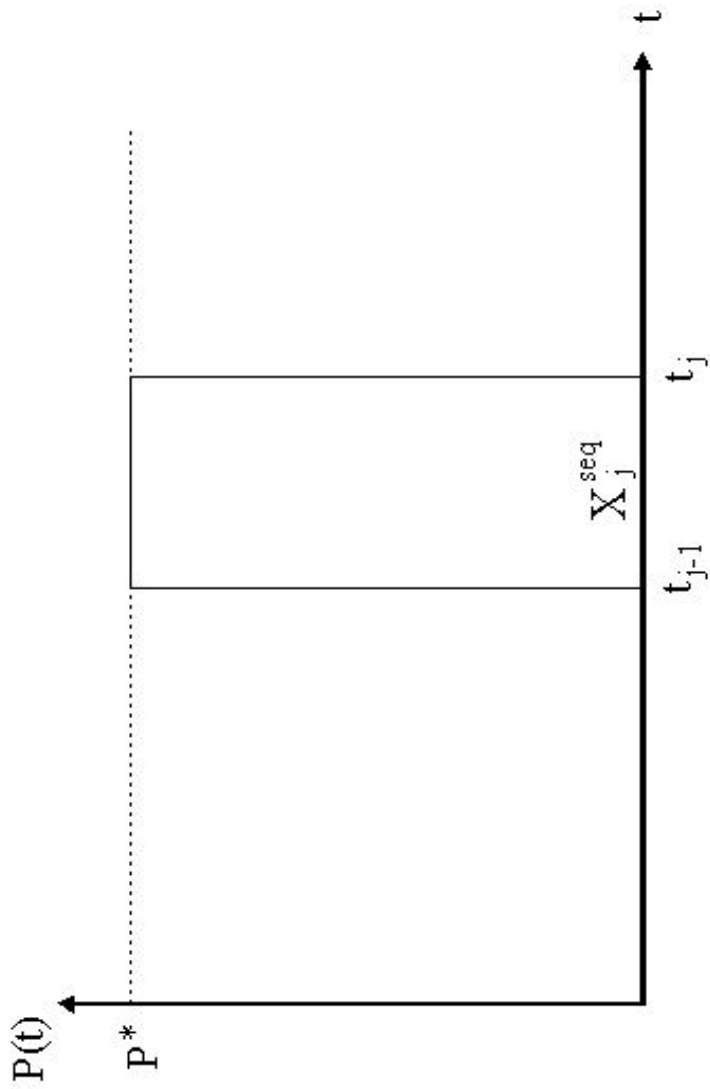


Рис. 1

вательная стратегия выполнения заданий.

По предположению, высказанному в §1, все задания имеют равный объём и, следовательно, одно и то же нормативное время решения, т.е. $\mathbf{T}_j = \mathbf{T}, j = \overline{1, n}$. Обозначим через X_j^{seq} время решения задачи из J' согласно стратегии $seq(S(J), E, t)$. Заметим, что $X_j^{seq} \equiv \mathbf{X}_j \forall j \in J'$, поскольку при использовании последовательной стратегии каждая задача из J' выполняется в монопольном режиме на всем ресурсе СВ-системы мощности V . Обозначим, как и ранее, в §1 через X^{seq} множество значений величин X_j^{seq} , т.е. $X^{seq} = \{X_j^{seq}, j = \overline{1, n}\}$.

После завершения выполнения последнего задания для любой из очередей из $\hat{S}(J')$ можно вычислить

суммарное время обработки всех заданий

$$U = \sum_{j=1}^n X_j^{seq},$$

среднее время решения одной задачи

$$\bar{X} = \frac{1}{n} U = \frac{1}{n} \sum_{j=1}^n X_j^{seq} \equiv \frac{1}{n} \sum_{j=1}^n \mathbf{X}_j.$$

Для каждого задания J_j из набора J' можно вычислить ξ_j — показатель относительной длительности исполнения

$$\xi_j^{seq} \equiv \xi_j = \frac{n \cdot \mathbf{X}_j}{\sum_{j=1}^n \mathbf{X}_j},$$

где $\mathbf{X}_j$ — абсолютное собственное время решения j -ой задачи из набора J' , $X_j^{seq} \equiv \mathbf{X}_j \forall j \in J'$.

Для конкретного набора J' в качестве очереди для обработки СВ-системой выберем некоторую перестановку

$\hat{s}_l = (J_a, \dots, J_z)$ элементов из J' . После завершения выполнения последнего задания из очереди $\hat{s}_l$ для каждого J_j можно вычислить время t_{jk}^{seq} его пребывания в системе, здесь j — номер задачи, k — её номер по порядку в очереди $\hat{s}_l$:

$$t_{jk}^{seq} = \sum_{i=1}^k X_i^{seq}.$$

Другими словами, время пребывания j -го задания в системе, при условии, что оно выполнялось k -м по порядку при использовании последовательной стратегии для зафиксированной очереди $\hat{s}_l$, равно сумме абсолютных времен всех выполненных перед ним заданий и абсолютной длительности этого задания.

Для каждой конкретной очереди $\hat{s}_l$ можно вычислить коэффициент относительного времени запаздывания j -го задания η_{jk}^{seq} , при условии, что оно выполнялось k -м по порядку (т.е. находилось на k -м месте в очереди):

$$\eta_{jk}^{seq} = \frac{t_{jk}^{seq}}{X_j},$$

где t_{jk}^{seq} — время пребывания j -й задачи в СВ-системе при условии, что она выполнялась k -ой по порядку в фиксированной последовательности $\hat{s}_l$. Ещё раз заметим, что значения X_j^{seq} и X_j совпадают для всех $j = \overline{1, n}$ из J .

Проведем умозрительный эксперимент и предположим, что для всех возможных очередей, т.е. для всех перестановок $\hat{s}_l$ из $\hat{S}(J')$, получены все соответствующие им наборы значений t_{jk}^{seq} .

Будем считать, что в ходе умозрительного эксперимента для каждого конкретного j -го задания из набора J' было получено $n!$ значений η_{jk}^{seq} . Рассмотрим следующие величины:

η_{j1}^{seq} — коэффициент относительного времени запаздывания j -го задания при использовании последовательной стратегии и

условии, что оно выполнялось первым по порядку. Из определения следует, что

$$\eta_{j1}^{seq} = \frac{t_{j1}^{seq}}{\mathbf{X}_j} = 1;$$

η_{jn}^{seq} — коэффициент относительного времени запаздывания j -го задания, при условии, что оно выполнялось последним, т.е. n -м по порядку:

$$\eta_{jn}^{seq} = \frac{t_{jn}^{seq}}{\mathbf{X}_j} = \frac{U}{X_j^{seq}} = \frac{n \cdot \bar{X}}{X_j^{seq}} = \frac{n}{\xi_j^{seq}}.$$

Указанные соотношения верны для всех номеров j .

На рис. 2 схематически изображена зависимость η_{jk}^{seq} , $k = \{1, n\}$, от ξ_j^{seq} для приведённых выше соотношений. При этом значения ξ_j^{seq} определены на интервале $[0; n]$, а значения η_{jn}^{seq} могут быть сколь угодно большими. Фактически, указанные линии характеризуют два граничных сценария размещения заданий в очереди: первый — когда задержки в получении решения не происходит, и второй — когда эта задержка максимальна.

По условию X_j^{seq} — реализация случайной величины, равномерно распределённой на отрезке $[0; \mathbf{T}]$, и $X_j^{seq} = \mathbf{X}_j \forall j = \overline{1, n}$. Для η_{jk}^{seq} можно выписать следующие равенства

$$\begin{aligned} \eta_{jk}^{seq} &= \frac{t_{jk}^{seq}}{\mathbf{X}_j} = \frac{k-1}{n-1} \cdot \frac{U}{\mathbf{X}_j} + \frac{n-k}{n-1} = \frac{k-1}{n-1} \cdot \frac{n \cdot \bar{X}}{\mathbf{X}_j} + \frac{n-k}{n-1} = \\ &= \frac{k-1}{n-1} \cdot \frac{n}{\xi_j^{seq}} + \frac{n-k}{n-1}. \end{aligned}$$

Тогда для $\eta_{j1}^{seq}, \dots, \eta_{jn}^{seq}$ среднее по всем вариантам расположения в последовательности значение для данного набора J'

$$\bar{\eta}_j^{seq} = \frac{n}{2\xi_j^{seq}} + \frac{1}{2}.$$

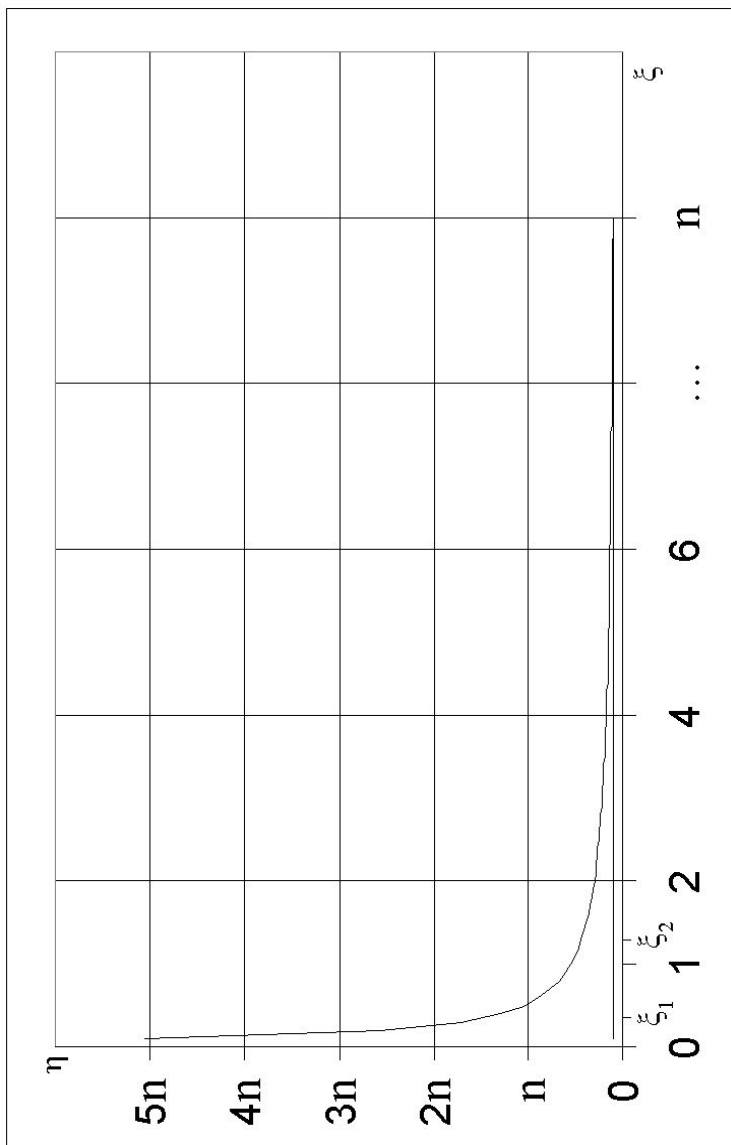


Рис. 2

Очевидно, что если $X_l^{seq} = \overline{X}$, $\xi_l^{seq} = 1$, то $\overline{\eta}_l^{seq} = \frac{n+1}{2}$.

На рис. 3 изображена диаграмма, схематично отражающая три возможных сценария размещения заданий из J' в очереди: в лучшем случае (когда задание выполняется первым), в худшем случае (когда задание выполняется последним) и в среднем. Полученные кривые могут служить оценкой для конкретной последовательности выполнения заданий из множества J' . В целом из диаграммы следует, что значения ξ_j^{seq} и η_j^{seq} связаны гиперболической зависимостью, и, соответственно, для ξ_j^{seq} из интервала $]0; 1[$ значения η_j^{seq} очень велики. Последнее на практике означает, что задачи с небольшой, меньшей среднего, собственной длительностью решения могут ожидать обработки достаточно долго по сравнению с абсолютным времени их счёта. В частности, для информационно-вычислительных систем можно говорить о потере актуальности решения содержательной задачи.

Если трактовать поставленную проблему как задачу о распределении вычислительных ресурсов между пользователями, то очевидно, что последовательная схема при неизвестной заранее длительности решения является дискриминирующей по отношению к заявкам, имеющим время решения меньше среднего. Действительно, в рамках высказанных предположений каждое задание из произвольного множества J имеет единственную уникальную характеристику — абсолютное время обработки X_j , однако её численное значение априори неизвестно, поэтому не может быть использовано для оптимизации распределения ресурсов. Случайный выбор последовательности выполнения заданий, не обусловленный объективными причинами, привносит дополнительную неопределённость, поскольку место в очереди на обработку конкретной задачи оказывается произвольным, что, в свою очередь, делает момент начала решения каждой задачи, кроме первой, случайной величиной,

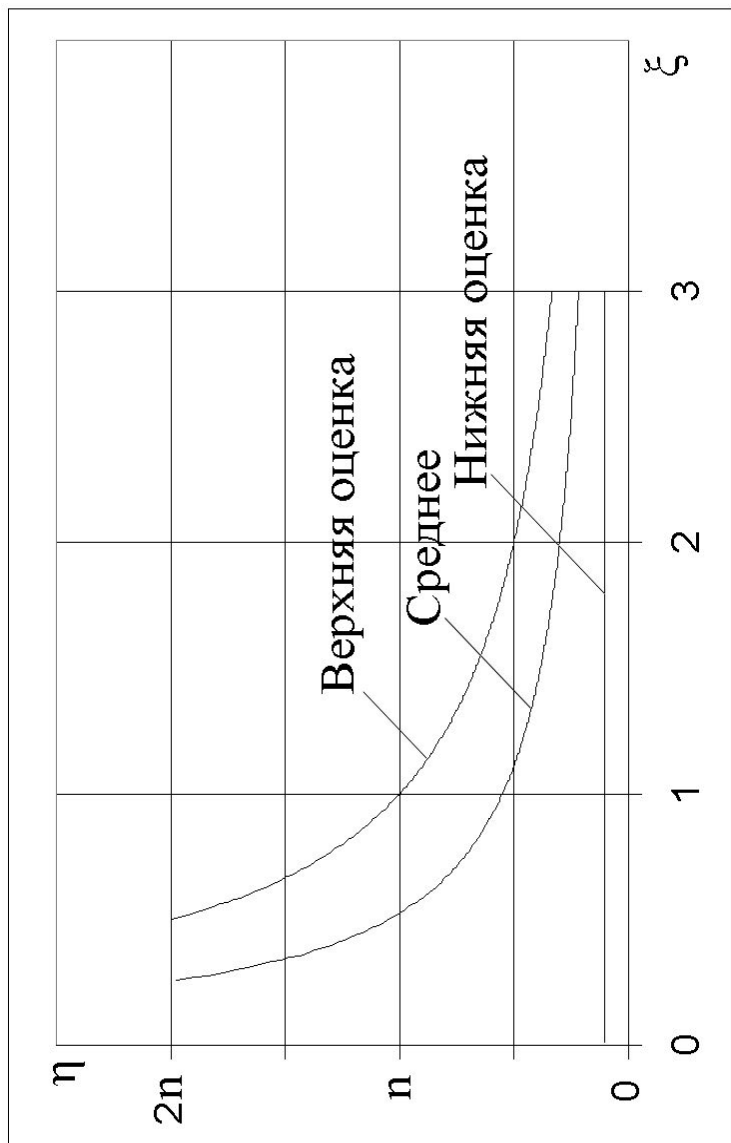


Рис. 3

зависящей от времени решения предыдущих задач.

Таким образом, последовательная схема не позволяет диспетчеру сделать достоверного прогноза о времени окончания конкретного задания из имеющегося фиксированного множества J' и не даёт возможности влиять на задержки в выполнении задач с коротким абсолютным временем обработки.

§3. Параллельное выполнение заданий

Рассмотрим следующую стратегию выполнения заданий, которую в дальнейшем будем называть параллельной. В начальный момент времени $t = 0$ начинается обработка всех имеющихся заданий множества J одновременно, каждому из них выделяется равное число ЕВ-модулей, т.е. $\forall l, l = \overline{1, n}, e_l = m/n$, и $E = (m/n, \dots, m/n)$. Напомним, что здесь m — общее число ЕВ-модулей.

Пусть в момент времени t_1 одна из задач, пусть j_1 -ая была завершена, тогда освободившийся ресурс разделяется между незавершенными заданиями $J \setminus \{J_{j_1}\}$ поровну, т.е. $\forall l = \overline{1, n}, l \neq j_1, e_l = m/(n-1), e_{j_1} = 0$, и $E = (m/(n-1), \dots, 0, \dots, m/(n-1))$.

Если в момент времени t_1 одновременно завершено k задач, пусть $j_1, j_2, \dots, j_k$, то освободившийся ресурс также разделяется между незавершенными заданиями $J \setminus \{J_{j_1}, \dots, J_{j_k}\}$ поровну, т.е. $\forall l = \overline{1, n}, l \neq j_1, l \neq j_2, \dots, l \neq j_k, e_l = m/(n-k), e_{j_1} = 0, \dots, e_{j_k} = 0$, и вектор E имеет на $j_1, j_2, \dots, j_k$ -х местах нули, т.е. $E = (m/(n-k), 0, \dots, m/(n-k), 0, \dots, m/(n-k))$.

Выделение всем незавершенным заданиям равного числа ЕВ-модулей объясняется тем, что все $J_j \in J$ имеют одинаковые входные параметры, а собственное время обработки становится известно только после завершения, поэтому такое распределение ресурса является ответом на указанную неопределённость и попыткой решить короткие задачи в первую очередь.

Перераспределение освободившегося ресурса производится в каждый t_k момент времени, когда завершилась хотя бы одна задача, до тех пор, пока множество J не будет исчерпано. Описанную стратегию будем называть параллельной стратегией (параллельной схемой обработки заданий) и обозначать $\pi^{par} = par(S(J), E, t)$.

Распределение ресурса для задания, которое было завершено в момент времени t_k при параллельной обработке множества J , схематично изображено на рис. 4.

Обозначим через t_1^{par} длительность пребывания в системе задания, вышедшего при параллельной обработке из системы первым, t_2^{par} — вторым по порядку, t_n^{par} — последним. Если в некоторый момент времени завершилось k задач одновременно, то порядковые номера для этих заданий можно присвоить, например, по возрастанию их порядковых номеров в множестве J . Обозначим через t^{par} множество значений величин t_j^{par} для фиксированного набора заданий J' , т.е. $t^{par} = \{t_j^{par}, j = \overline{1, n}\}$.

При использовании параллельной стратегии имеющийся ресурс изначально был поделен между всеми заданиями из J поровну, поэтому первым завершилось задание, имеющее наименьшее абсолютное время выполнения $\mathbf{X}_1^* = \min_n \{\mathbf{X}_j | \mathbf{X}_j \in \mathbf{X}\}$. Напомним, что каждый элемент $\mathbf{X}_j$ из $\mathbf{X}$ — время решения j -й задачи СВ-системой в монопольном режиме, а для выполнения каждого задания при параллельной обработке было задействовано V/n всех ЕВ-модулей. Поэтому на выполнение первого завершеного задания пришлось затратить в n раз больше времени, чем при последовательной обработке, т.е.

$$t_1^{par} = n \cdot \mathbf{X}_1^*.$$

Таким образом, при использовании параллельной стратегии первой систему покидает задача, которая имеет наименьшее абсолютное время решения, а её реальное время пребывания оказывается в n раз больше, чем её абсолютное время.

Для удобства дальнейшего изложения переупорядочим множество $\mathbf{X}$, соответствующее набору J' , по неубыванию его элементов и перенумеруем следующим образом:

$$\mathbf{X}_1^* = \min_j \{\mathbf{X}_j | \mathbf{X}_j \in \mathbf{X}\}, \mathbf{X}_2^* = \min_j \{\mathbf{X}_j | \mathbf{X}_j \in \mathbf{X} \setminus \{\mathbf{X}_1^*\}\}, \dots,$$

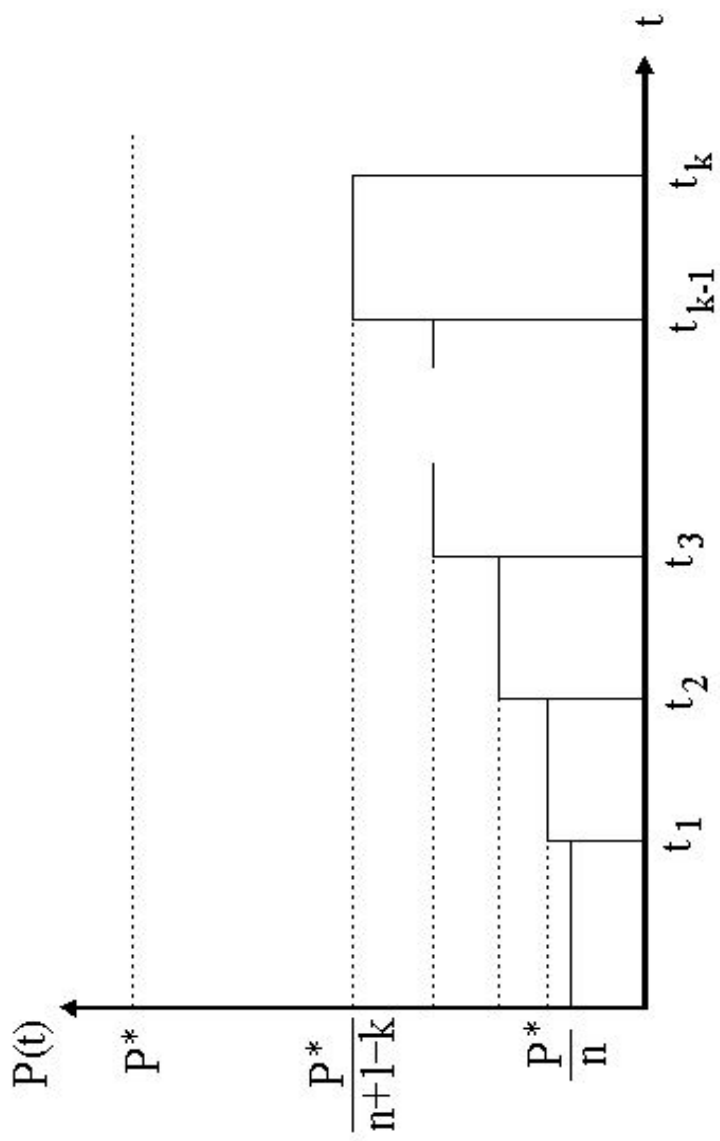


Рис. 4

$$\mathbf{X}_n^* = \max_j \{\mathbf{X}_j \mid \mathbf{X}_j \in \mathbf{X}\}$$

Полученное множество обозначим $\mathbf{X}^* = \{\mathbf{X}_1^*, \dots, \mathbf{X}_j^*, \dots, \mathbf{X}_n^*\}$,
для всех элементов

$$\mathbf{X}_1^* \leq \mathbf{X}_2^* \leq \dots \leq \mathbf{X}_i^* \leq \mathbf{X}_{i+1}^* \leq \dots \leq \mathbf{X}_n^*.$$

Тогда

$$t_1^{par} = n\mathbf{X}_1^*,$$

откуда следует, что

$$\eta_{11}^{par} = \frac{t_1^{par}}{\mathbf{X}_1^*} = n.$$

Это соответствует максимальному значению среди всех η_{jk}^{par} .
Далее второй нижний индекс у величины η_{jk}^{par} будем опускать,
поскольку при параллельной обработке задания выходят из системы
в строгом порядке, определённом последовательностью $\mathbf{X}^*$.

Для задачи, имеющей второе по величине абсолютное время
решения, при параллельной обработке

$$\begin{aligned} t_2^{par} &= n\mathbf{X}_1^* + (n-1)(\mathbf{X}_2^* - \mathbf{X}_1^*) = (n-1)\mathbf{X}_2^* + \mathbf{X}_1^* = \\ &= (n-2)\mathbf{X}_2^* + (\mathbf{X}_1^* + \mathbf{X}_2^*), \\ \eta_2^{par} &= \frac{t_2^{par}}{\mathbf{X}_2^*} = (n-1) + \frac{\mathbf{X}_1^*}{\mathbf{X}_2^*} = n - (1 - \frac{\mathbf{X}_1^*}{\mathbf{X}_2^*}). \end{aligned}$$

Для задачи, собственное время решения которой оказалось
максимальным среди всех заданий

$$t_n^{par} = U = n \cdot \bar{X},$$

$$\eta_n^{par} = \frac{t_n^{par}}{\mathbf{X}_n^*} = \frac{U}{\mathbf{X}_n^*} = \frac{n \cdot \bar{X}}{\mathbf{X}_n^*} = n - n(1 - \frac{\bar{X}}{\mathbf{X}_n^*}).$$

В общем случае

$$t_k^{par} = (n - k)\mathbf{X}_k^* + \sum_{i=1}^k \mathbf{X}_i^*, \quad (2)$$

$$\eta_k^{par} = \frac{t_k^{par}}{n\mathbf{X}_k^*} = 1 - \frac{1}{n} \sum_{i=1}^k (1 - \frac{\mathbf{X}_i^*}{\mathbf{X}_k^*}),$$

при этом

$$\bar{\eta}_1^{par} \leq \bar{\eta}_1^{seq}, \dots, \bar{\eta}_n^{seq} \leq \bar{\eta}_n^{par}.$$

Таким образом, оказалось, что $\xi_n^{par} \leq n$ для любого $j = \overline{1, n}$.

Результаты умозрительного эксперимента для фиксированного набора J' , содержащего n заданий, могут быть изображены в виде набора точек в координатах $\langle \xi, \eta \rangle$. Предположим, что проведено имитационное моделирование для большого числа различных наборов J и результаты всех экспериментов — точки — соединены для наглядности линиями. На рис. 5 помещены кривые, описывающие средние значения переменной η для последовательной и параллельной схем выполнения, которые отражают результат такого эксперимента.

Из рис. 5 видно, что график для средних значений $\bar{\eta}^{seq}$ последовательной схемы и график η^{par} параллельной схемы пересекаются. Это означает, что при параллельной схеме выполнения короткие задания, с малыми значениями ξ , имеют коэффициент относительной задержки выполнения меньше, чем при последовательной схеме, поскольку выполняются в первую очередь.

Вернёмся к последовательной схеме выполнения и рассмотрим очередь-перестановку $\hat{s}^*$, согласно которой первой выполняется задача, имеющая абсолютное время решения $\mathbf{X}_1^*$, второй

— $\mathbf{X}_2^*$, и т.д. Таким образом, порядок выполнения для очереди $\hat{s}^*$ совпадает с порядком решения задач при параллельной схеме. Вычислим t_k^{seq} для очереди $\hat{s}^*$ и $k = \overline{1, n}$:

$$t_k^{seq} = \sum_{i=1}^k \mathbf{X}_i^*.$$

Вычислим значение $\Delta t_k = t_k^{par} - t_k^{seq}$ для рассматриваемой очереди $\hat{s}^*$. Согласно (2)

$$\Delta t_k = t_k^{par} - t_k^{seq} = (n - k)\mathbf{X}_k^*.$$

Тогда величина

$$\delta_k = \frac{\Delta t_k}{\mathbf{X}_k^*} = (n - k) \quad (3)$$

показывает относительную задержку выполнения k -й задачи при параллельной схеме по сравнению с очередью $\hat{s}^*$ последовательной схемы. Величина δ_k не зависит от значений $\mathbf{X}_k^*$, и, следовательно, от закона распределения X_k . Таким образом, δ_k в определённом смысле является гарантированной динамической оценкой.

Будем говорить, что очередь $\hat{s}^*$ является *нижней огибающей* всех очередей-перестановок из $\hat{S}(J')$ для набора J' . Действительно, вычислим минимальное значение времени пребывания t_i^* в системе любого задания, которое покидает СВ-систему i -м по порядку для всех возможных очередей-перестановок при последовательной стратегии обслуживания. Очевидно, что

$$t_1^* = \mathbf{X}_1^*, t_2^* = \mathbf{X}_1^* + \mathbf{X}_2^*, \dots,$$

$$t_k^* = \sum_{i=1}^k \mathbf{X}_i^* \quad \forall \quad k = \overline{1, n}.$$

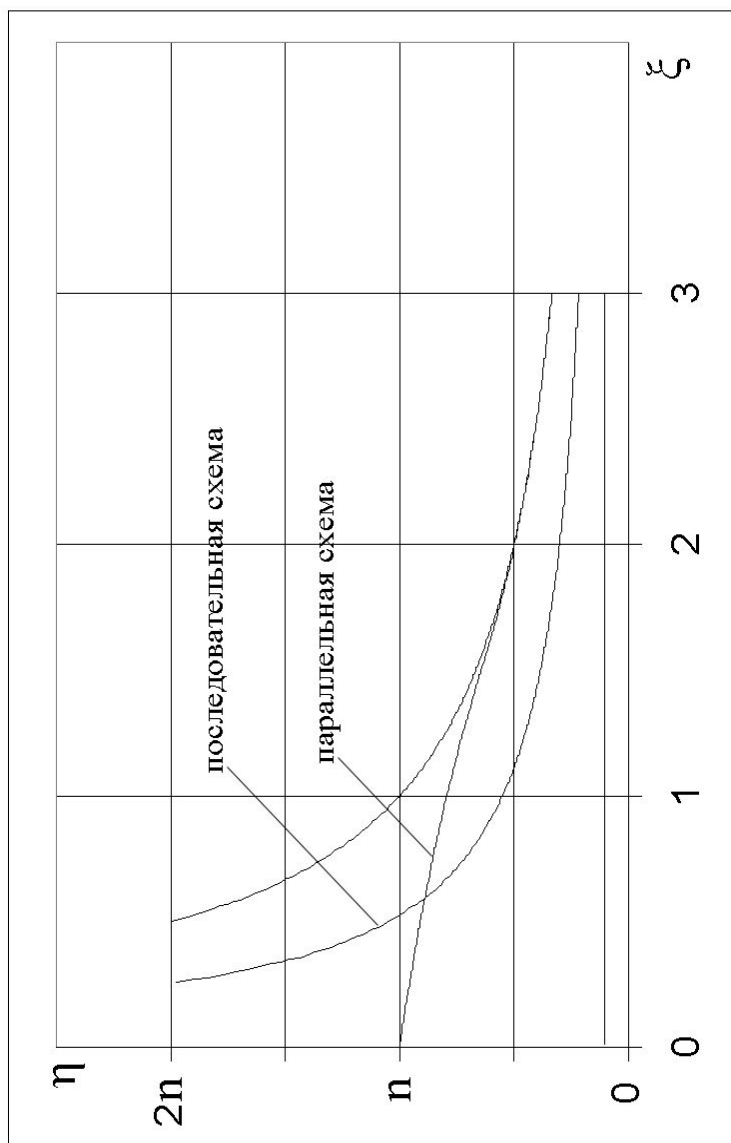


Рис. 5

Отсюда следует, что

$$\bar{t}^* = \frac{1}{n} \sum_{i=1}^n t_i^*$$

является минимальным средним временем пребывания заданий в СВ-системе при последовательной схеме выполнения.

Таким образом, с одной стороны, процесс выполнения заданий из множества J' с помощью очереди $\hat{s}^*$ можно описать последовательностью минимальных (неулучшаемых) значений $\{t_1^*, t_2^*, \dots, t_n^*\}$ времени выхода из системы k -й по порядку задачи. С другой стороны, параллельная схема позволяет получить тот же порядок выхода заданий с гарантированной динамической оценкой (см. формулу (3)) времен относительно наилучшей из возможных траекторий, задаваемых в координатах $\langle t_k, k \rangle$.

Таким образом, параллельная схема выполнения заданий обладает следующими свойствами:

- автоматически обеспечивает завершение в первую очередь заданий, имеющих минимальное абсолютное время решения, хотя реальная длительность обработки заданий становится известна только после их выполнения;
- фактически задания оказываются упорядочены в порядке неубывания длительности работ;
- в ходе выполнения заданий происходит сортировка имеющегося множества J таким образом, что более "длинные" задания завершаются последними, а число ЕВ-модулей, выделяемое им для обработки, монотонно возрастает;
- позволяет найти оптимальное распределение ресурсов, при котором достигается

$$\min_{\pi(S(J), E, t)} \max_{1 \leq j \leq n} \left(\frac{t_j^\pi}{\mathbf{X}_j^*} \right),$$

который оказывается равным n . Заметим, что ни для какой другой стратегии эта величина не может быть меньше n ;
— для совокупности заданий из J позволяет получить в динамике гарантированные оценки "уклонения" времени выполнения от минимально возможных неулучшаемых значений.

§4. Вычислительный эксперимент

Начнём параграф с цитаты из [13]:— Сорок два?! - завопил Лункуал. - И это все, что ты можешь нам сказать после семи с половиной миллионов лет работы? — Я убежден в правильности ответа, - холодно отрезал компьютер. — По правде говоря, — прибавил он, смягчившись, - дело, я думаю, в том, что вы никогда, собственно, не задумывались, в чем состоит этот вопрос... Вот когда будете знать вопрос, то поймете и ответ".

Для исследования модели, описанной в пункте 2, были проведены следующие имитационные вычислительные эксперименты. Набор заданий генерировался случайным образом. Далее, в первом случае, моделировалась последовательная стратегия: задания выполнялись последовательно одно за другим в монопольном режиме на всем вычислительном ресурсе. Во втором случае моделировался параллельный режим: на выполнение устанавливался весь пакет одновременно, т.е. имеющиеся задания выполнялись параллельно. При этом мощность СВ-системы первоначально распределялась между всеми задачами поровну, в дальнейшем освободившиеся ресурсы делились между оставшимися заданиями также поровну. Пакеты формировались как из задач со случайным временем окончания решения, равномерно распределённым на заданном временном интервале, так и из заданий с фиксированным временем счёта. Целью эксперимента являлось сравнение результатов обработки пакетов задач, обладающих различными характеристиками, при параллельной и последовательной стратегиях.

Перейдем к более детальному описанию имитационной модели, использованной в экспериментах. Производительность вычислительной системы принималась за единицу. При последовательном выполнении новое задание ставилось на обработку в момент завершения предыдущего. В случае параллельного выполнения считалось, что ресурс вычислительной системы в на-

чале разделен поровну, и в момент завершения задания ресурс распределяется так же между оставшимися незавершенными. Предполагалось, что все операции — постановка на выполнение, завершение задания и перераспределение ресурсов производятся мгновенно без потери времени. Таким образом, во всех экспериментах рассматривались две стратегии — последовательное и параллельное выполнение с равномерным распределением освобождающихся ресурсов между незавершенными заданиями.

Для каждой модели генерировалось 10^6 случайных реализаций наборов данных, вычислялись выборочные средние и среднеквадратичные отклонения исследуемых величин. Результаты счета сравнивались по следующим показателям: среднее время завершения i -го задания, их среднеквадратичное отклонение и др. Результаты приводятся ниже в виде графиков и таблиц.

Рассматривалось четыре типа наборов заданий. Каждый набор содержал 12 заданий, максимальное время обработки каждого задания равнялось 10 часам. Для наборов первого типа время завершения каждого задания считалось случайной величиной, равномерно распределённой на отрезке от 0 до 10 часов, при этом все задачи набора имели решение, т.е. в каждой урне содержался чёрный шар. Для наборов второго типа считалось, что ровно 2 задачи из 12-ти не имеют решения, т.е. две произвольные урны из двенадцати не содержат чёрного шара, и белые перебираются до последнего шара. Длительность решения двух этих задач составила 10 часов. Длительность обработки каждого из 10-ти оставшихся заданий полагалась случайной величиной, равномерно распределённой на отрезке $[0; 10]$. Таким образом, для наборов второго типа соотношение числа задач, имеющих решение, и заданий, которые выполняются до конца, равно 10:2. Для наборов третьего и четвёртого

типов аналогичные соотношения задавались равными 9:3 и 8:4, соответственно.

Для каждого набора рассматривались все возможные перестановки заданий внутри набора. Для каждого набора и каждой перестановки заданий внутри набора рассматривались две стратегии их выполнения: последовательная и параллельная. По мере завершения заданий им присваивались порядковые номера. Было сгенерировано 10^6 наборов, для каждого из которых рассматривалось $12!$ перестановок, и для каждой перестановки вычислялись различные характеристики, например, время окончания задания. Затем для всех перестановок внутри набора определялись промежуточные результаты, которые затем усреднялись по всем наборам. Заметим, что при параллельной стратегии порядок задач в наборе не влияет на порядок завершения задач и, следовательно, на значения соответствующих показателей, кроме того, момент окончания обработки всех заданий из пакета для обеих стратегий совпадает.

Для каждого набора и всех задач, имеющих решение, вычислялись: выборочное среднее время завершения, выборочное среднеквадратичное отклонение и выборочный коэффициент вариации. Перечисленные величины усреднялись по всем наборам. Полученные результаты представлены в табл. 1.

Для пакетов всех типов среднее время пребывания задания в системе при последовательном выполнении оказывается меньше, чем при параллельном. Однако, если при использовании последовательной стратегии задания покидают систему в произвольном порядке в среднем через одинаковые интервалы, то при параллельной обработке короткие задачи завершаются в первую очередь, моменты окончания сдвинуты к концу временного интервала. При этом задания покидают систему через отрезки времени, величины которых в среднем сокращаются.

Для всех типов пакетов при использовании параллельной

Табл. 1

Тип пакета		1	2	3	4
Соотношение есть решение : нет решения		12:0	10:2	9:3	8:4
Среднее время пробы	последовательная стратегия	32.5	37.5	40.0	42.5
	параллельная стратегия	41.7	45	46.7	48.3
Среднеквадратичное отклонение	последовательная стратегия	17.5	20	21.3	22.4
	параллельная стратегия	16.9	19.2	20.3	21.4
Коэффициент вариации	последовательная стратегия	0.55	0.55	0.55	0.55
	параллельная стратегия	0.42	0.45	0.46	0.47

стратегии выполнения вычислялись отдельно положительные и отрицательные значения отклонения времени окончания от выборочного среднего. По ним определялись выборочные среднеквадратичные отклонения по положительным и отрицательным значениям отдельно. Кроме того, производился подсчёт числа положительных и отрицательных отклонений в выборке. Все эти величины усреднялись по всем наборам задач. Полученные результаты помещены в табл. 2, где через σ^+ обозначено усреднённое среднеквадратичное значение положительных отклонений времени окончания задач, имеющих решение, от выборочного среднего, K^+ — усреднённое число положительных отклонений в выборке, σ^- , K^- — то же самое для отрицательных отклонений от выборочного среднего.

Анализ результатов табл. 2 показывает, что всегда при использовании параллельной стратегии $\sigma^+ < \sigma^-$, т.е. интервал времени между моментами окончания задач становится все меньше по мере их завершения. Кроме того, большее число задач выполняется позже среднего времени решения заданий из пакета: $K^+ > K^-$.

Табл. 2

Тип пакета, соотношение есть решение : нет решения	σ^+	σ^-	K^+	K^-
1, 12:0	13.8	20.9	6.9	5.1
2, 10:2	16.7	22.6	5.6	4.4
3, 9:3	18.1	23.5	4.9	4.1
4, 8:4	19.5	24.3	4.3	3.7

Для каждого набора J' и каждой стратегии, по всем возможным перестановкам задач внутри данного набора J' , определялись следующие характеристики:

- среднее время выполнения последней задачи, имеющей решение, для данного набора;
- разница во времени между выходом последней задачи, име-

ющей решение в данном наборе, при последовательной и параллельной обработке т.е. $\delta = t_k^{seq} - t_k^{par}$, и соответствующие средние квадратичные по перестановкам: положительные значения δ^+ и отрицательные значения δ^- для каждого;
— среднее значение η_i для каждого значения ξ_i в наборе, для задач, имеющих решение.

Полученные значения усреднялись по всем наборам. После усреднения η по всем наборам строился поточечный график зависимости η от ξ . Остальные характеристики выражались числовыми значениями: среднее время окончания последней задачи, имеющей решение в данном эксперименте, — k числами ($i = \overline{1, k}$), а среднеквадратичные значения положительных и отрицательных δ — каждое одним числом.

Двухэтапный процесс вычисления среднего (сначала по перестановкам внутри набора, затем по наборам) для средних времен окончания i -й задачи, имеющей решение, и для зависимости η от ξ приводит к тем же результатам, что и простое усреднение по наборам, т.е. когда задачи в наборе расставляются в одном единственном случайном порядке. Последнее утверждение следует с одной стороны из того, что любая перестановка внутри набора может быть получена при создании единственного порядка задач в наборе, и все перестановки имеют равную вероятность; с другой стороны из того, что все значения при двухэтапном усреднении входят в результат с одинаковым весовым коэффициентом.

Однако среднеквадратичные значения положительных и отрицательных δ при двухэтапном усреднении могут отличаться от результатов одноэтапного усреднения поскольку, во-первых, распределение δ отличается для разных наборов; во-вторых, число положительных и отрицательных δ в разных наборах различно, поэтому отдельные значения входят в результат с разными весовыми коэффициентами.

Для уменьшения объёма вычислений средние статистические характеристики по всем возможным перестановкам задач внутри набора вычислялись аналитически. Для получения обобщённых средних значений по наборам задач создавалось 10^6 наборов. Основные результаты экспериментов для фиксированного набора заданий собраны в табл. 3. В ней для выборок, в которых время выполнения последней задачи при последовательной схеме оказалось больше, чем для параллельной, через $Part^+$ обозначена доля положительных отклонений, через $SQRt^+$ — среднеквадратичное значение превышения времени. Через $Part^-$ обозначена доля отрицательных отклонений, через $SQRt^-$ — среднеквадратичное значение отрицательных отклонений для времени для выборок, в которых время выполнения последней задачи при последовательной схеме оказалось меньше, чем при параллельной.

Для примера рассмотрим результаты, полученные для пакета заданий третьего типа. Время окончания девятой задачи, имеющей решение, при использовании параллельной стратегии не зависит от порядка задач и равно

$$\sum_{i=1}^9 X_i + 3X_9.$$

Для последовательной стратегии время окончания 9-й задачи, имеющей решение, является случайной величиной, которая может принимать 4 значения:

$$\begin{aligned} &\sum_{i=1}^9 X_i \text{ с вероятностью } 1/220, \\ &\sum_{i=1}^9 X_i + 10 \text{ с вероятностью } 9/220, \\ &\sum_{i=1}^9 X_i + 20 \text{ с вероятностью } 45/220, \\ &\sum_{i=1}^9 X_i + 30 \text{ с вероятностью } 165/220. \end{aligned}$$

Следовательно, среднее значение составит $\sum_{i=1}^9 X_i + 10 \cdot 21/8$.

Нетрудно убедиться, что для любого конкретного набора J' средние значения времен окончания девятой задачи при ис-

Табл. 3

Тип пакета	1	2	3	4
Соотношение есть решение : нет решения	12:0	10:2	9:3	8:4
Среднее время выполнения последней задачи, имеющей решение	π^{seq} 60.0	68.2	72.0	75.6
	π^{par} 60.0	68.2	72.0	75.6
Положительные отклонения. Для выборки, в которых время выполнения последней задачи при последовательной схеме оказалось больше, чем при параллельной	$SQRt^{+}$ —	1.8	3.0	4.3
	$Part^{+}$ —	83%	76%	69%
Отрицательные отклонения. Для выборки, в которых время выполнения последней задачи при последовательной схеме оказалось меньше, чем при параллельной	$SQRt^{-}$ —	9.6	10.6	11.5
	$Part^{-}$ —	17%	24%	31%

пользовании разных стратегий могут различаться. Однако, после усреднения по всем наборам получились одинаковые величины — 72.0.

При вычислении средних квадратичных значений положительных и отрицательных δ для одного набора задач не нужно перебирать все перестановки задач внутри набора, а достаточно определить положение величины $3 \cdot X_9$ среди чисел 0, 10, 20, 30, и взять четыре разности с нужным знаком и с соответствующей вероятностью.

В результате статистического моделирования среднее значение среднеквадратичного положительного значения δ по всем наборам задач в эксперименте 3 оказалось равным 2.97, а значение среднеквадратичного отрицательного значения δ — 10.59, почти в 3 раза больше. При этом в 76% случаев время последовательной обработки больше, чем параллельной, а в 24% случаев — меньше. Последнее означает, что распределение δ является сильно несимметричным. Другими словами, положительные значения δ встречаются часто, но имеют относительно небольшую величину, а отрицательные значения δ встречаются редко, но больше по величине.

Это результат предсказуем, поскольку его можно получить аналитически. Действительно, подставим среднее значение $X_9 = 10 \cdot 7/8$ в формулу для среднего времени окончания 9-й задачи при параллельной стратегии $\sum_{i=1}^9 X_i + 3 \cdot X_9$. Получим: $3 \cdot X_9 = 26.25$. Сравним эту величину с рядом чисел 0, 10, 20, 30, которые получены для последовательной стратегии. Очевидно, что среднее значение $3 \cdot X_9$ немного меньше значения 30 (наиболее вероятного значения) и намного больше значений 0, 10, 20 вероятность которых меньше. Учитывая, что $\delta = t_9^{seq} - t_9^{par}$, можно сказать, что 9-я задача при последовательной стратегии довольно часто оканчивается немного позже, чем при параллельной, но если оканчивается раньше,

что происходит редко, то намного. В среднем, 9-я задача при обеих стратегиях оканчивается в одно время, через 72 часа.

Из анализа табл. 3 следует, что в большинстве случаев последняя задача, имеющая решение в заданном пакете заданий, решается раньше при параллельной обработке, чем при последовательной.

Диаграммы, позволяющие оценить среднее время окончания i -й задачи, имеющей решение для всех типов пакетов помещены, соответственно, на рис. 6, 8, 10, 12, графики зависимости среднего значения η от ξ — на рис. 7, 9, 11, 13. При проведении вычислительных экспериментов были получены данные, позволяющие анализировать точки пересечения различных графиков-диаграмм с рис. 7, 9, 11, 13.

Далее будем называть:

- линией $\langle \eta, \xi \rangle^{par}$ диаграмму (график) значений $\langle \eta, \xi \rangle$, полученных для параллельной стратегии выполнения заданий из набора J' ;
- кривой $\langle \eta, \xi \rangle_{mid}^{seq}$ диаграмму (график) средних значений $\langle \eta, \xi \rangle$, полученных на перестановках для последовательной стратегии выполнения заданий из набора J' ;

Как и ранее, для каждого конкретного набора J' в начале рассматривались все возможные перестановки заданий внутри набора и на основании полученных данных строились соответствующие им линия $\langle \eta, \xi \rangle^{par}$ и кривая $\langle \eta, \xi \rangle_{mid}^{seq}$. Кроме этого, определялись значения η^* и ξ^* — показателя абсолютного времени исполнения и коэффициента относительного запаздывания в точке пересечения линии $\langle \eta, \xi \rangle^{par}$ с кривой $\langle \eta, \xi \rangle_{mid}^{seq}$, соответственно, а также $\bar{\xi}$ — среднее значение ξ для всех задач, имеющих решение в данном наборе, и соответствующие значения ξ^{par} и ξ_{mid}^{seq} .

Далее вычислялась разность $\Delta\xi = \xi^* - \bar{\xi}$, она может быть

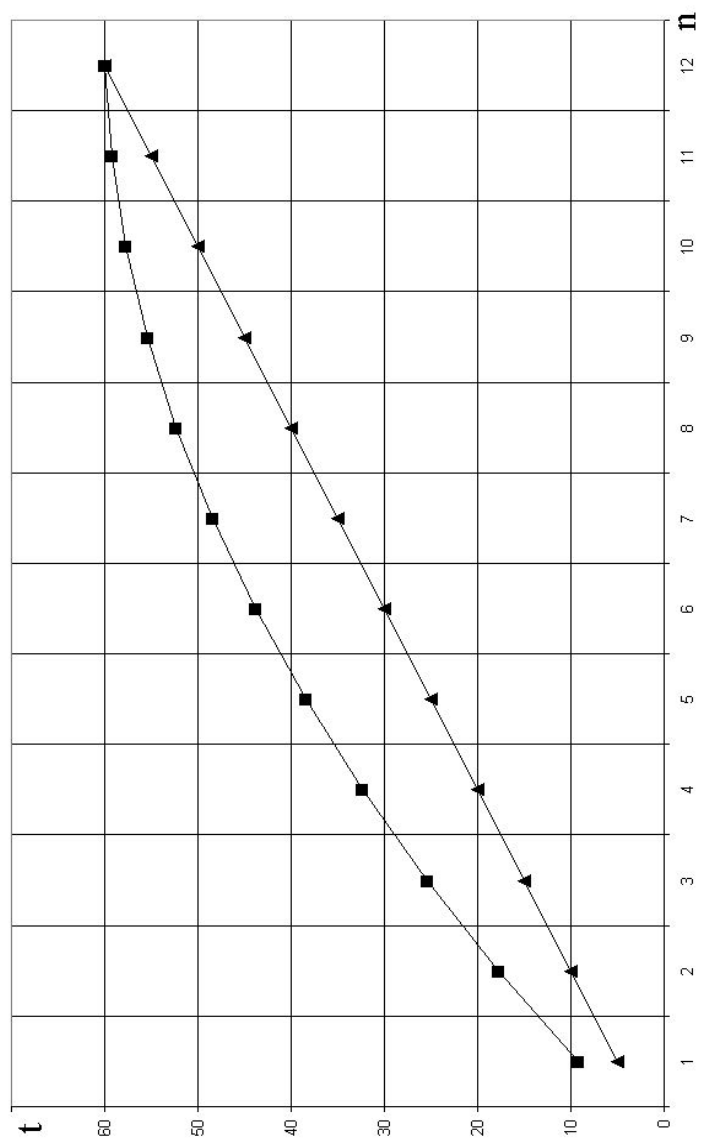


Рис. 6

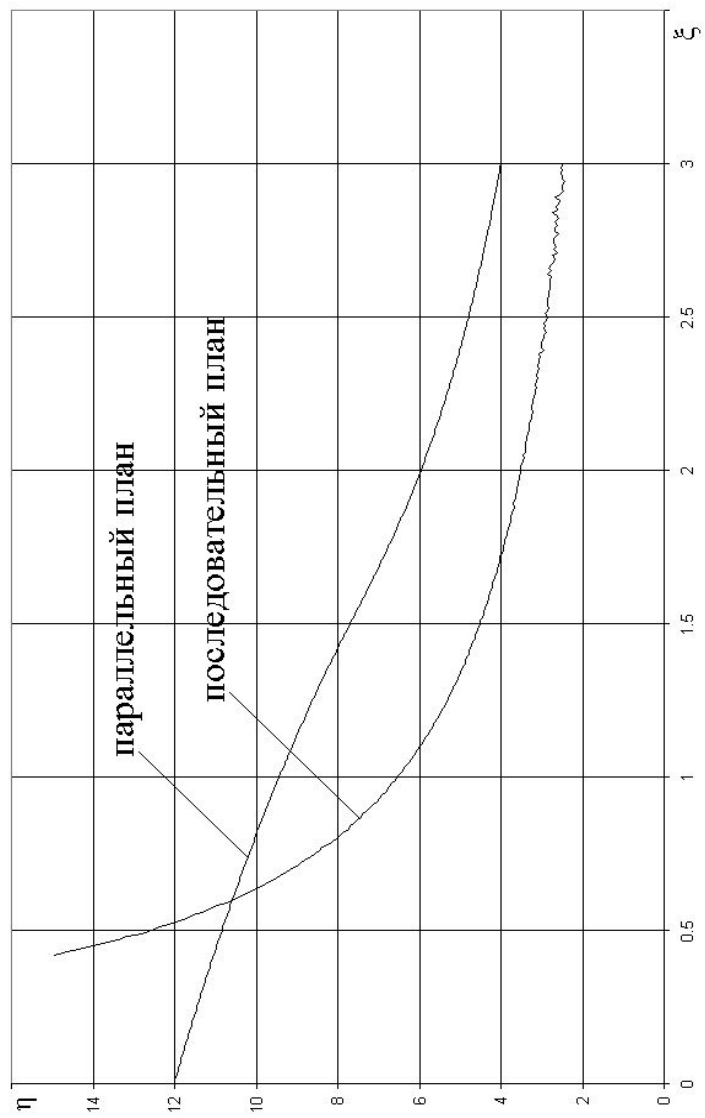


Рис. 7

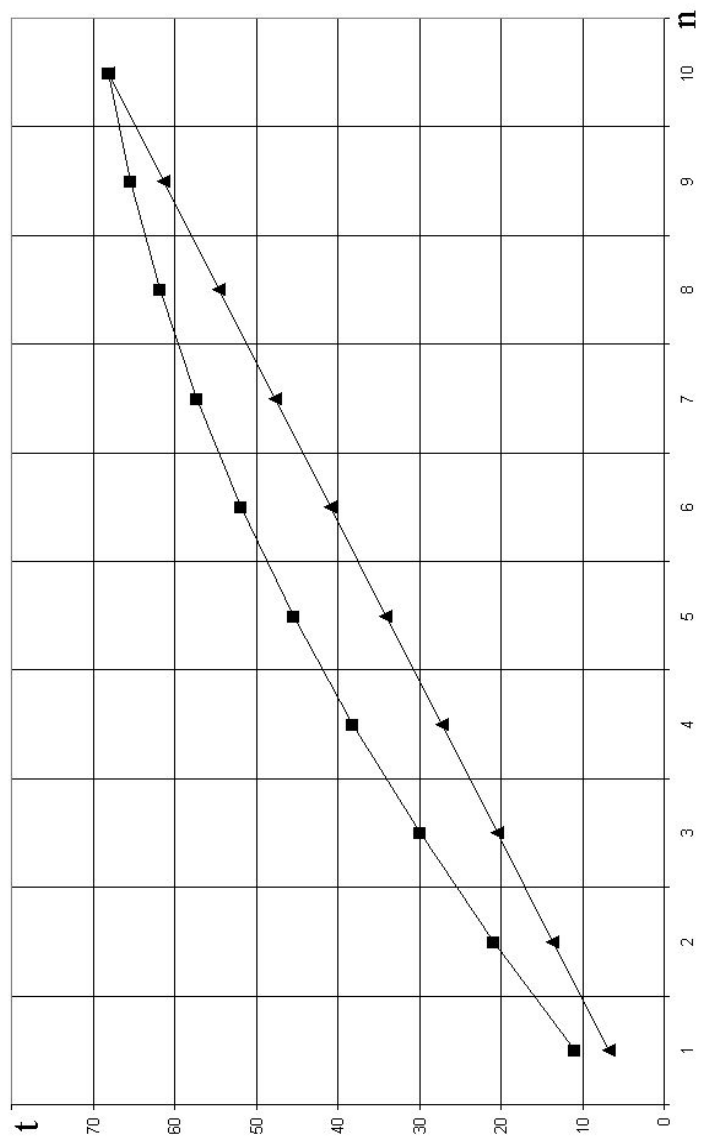


Рис. 8

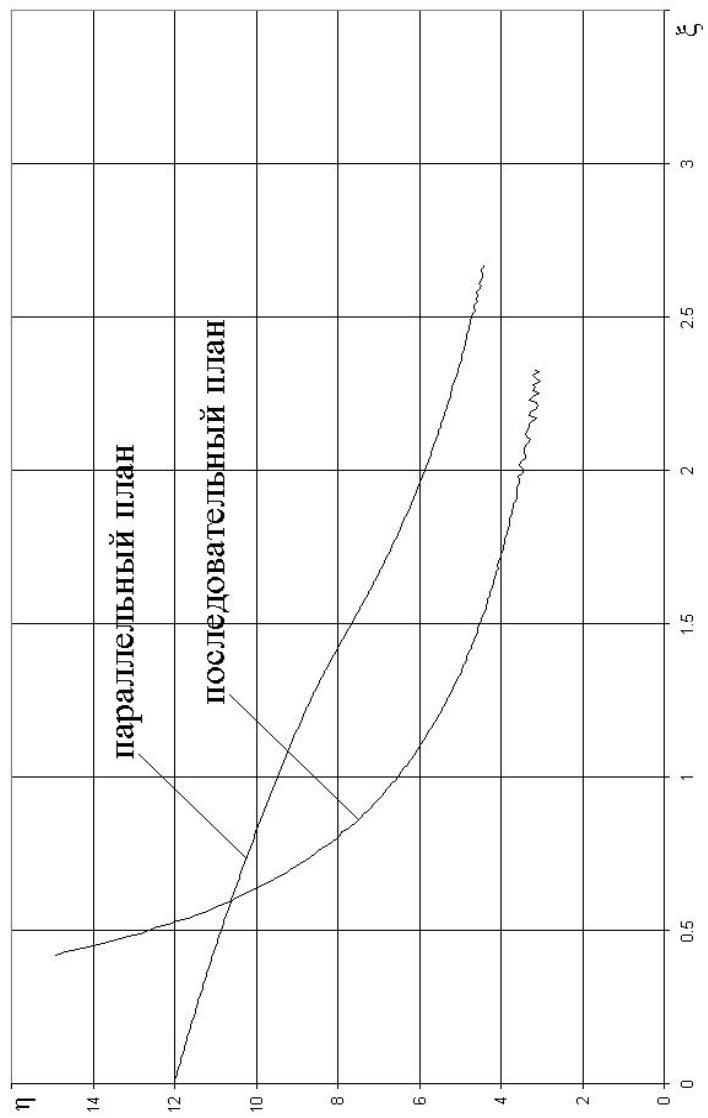


Рис. 9

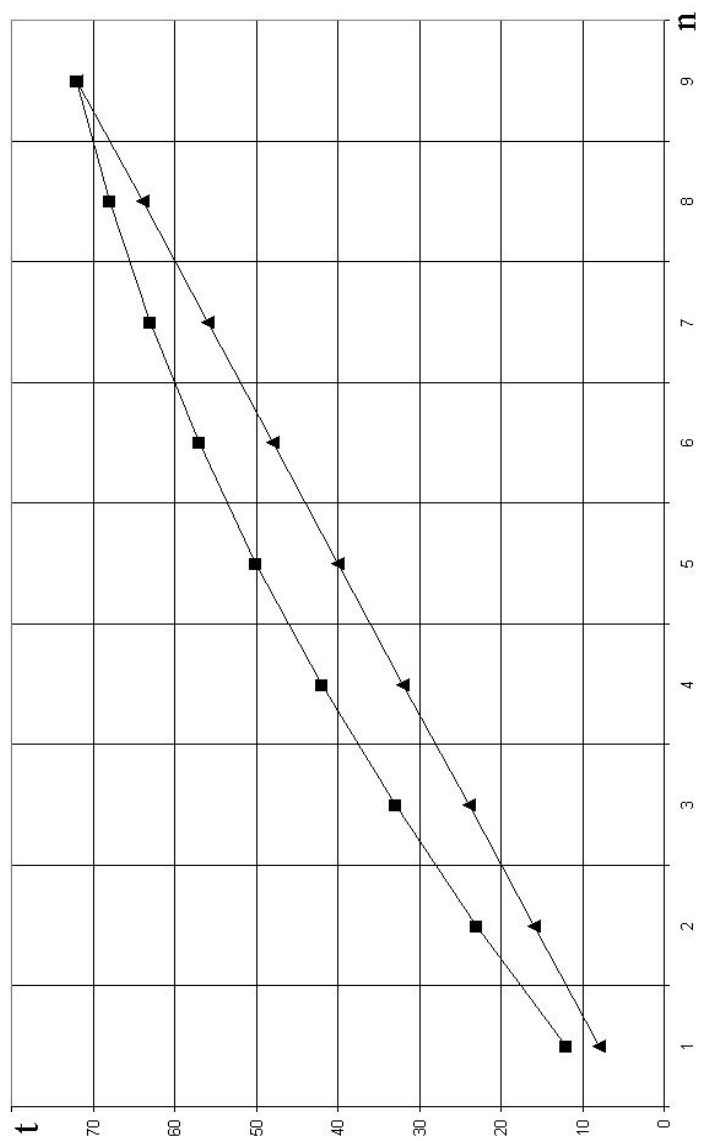


Рис. 10

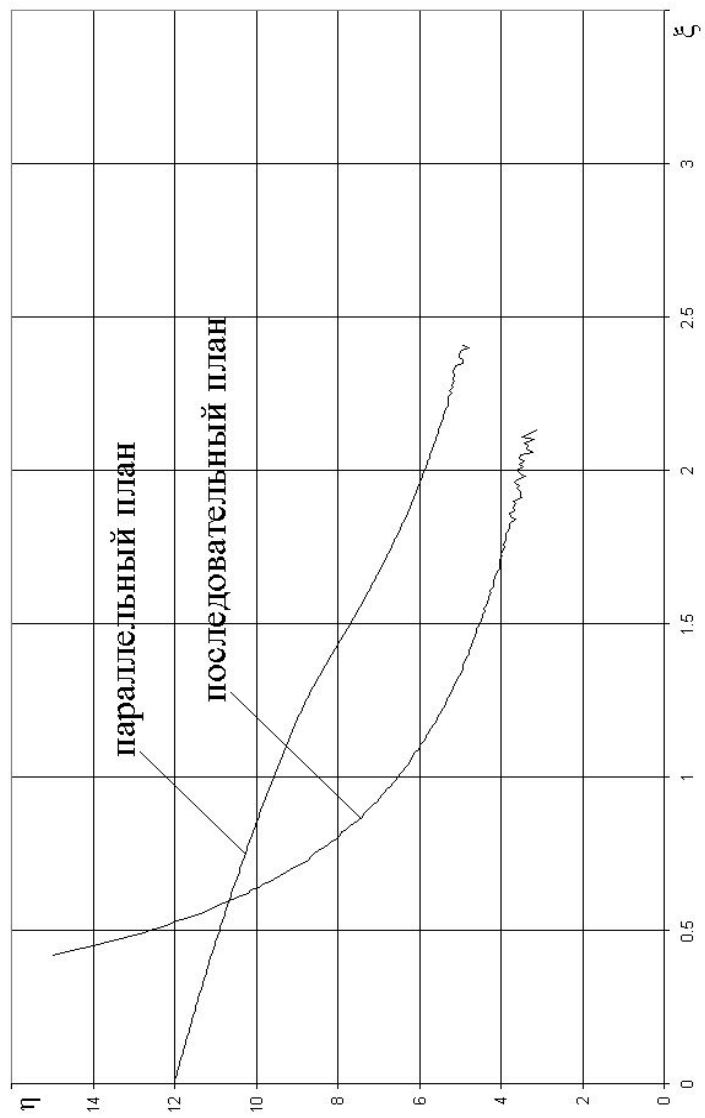


Рис. 11

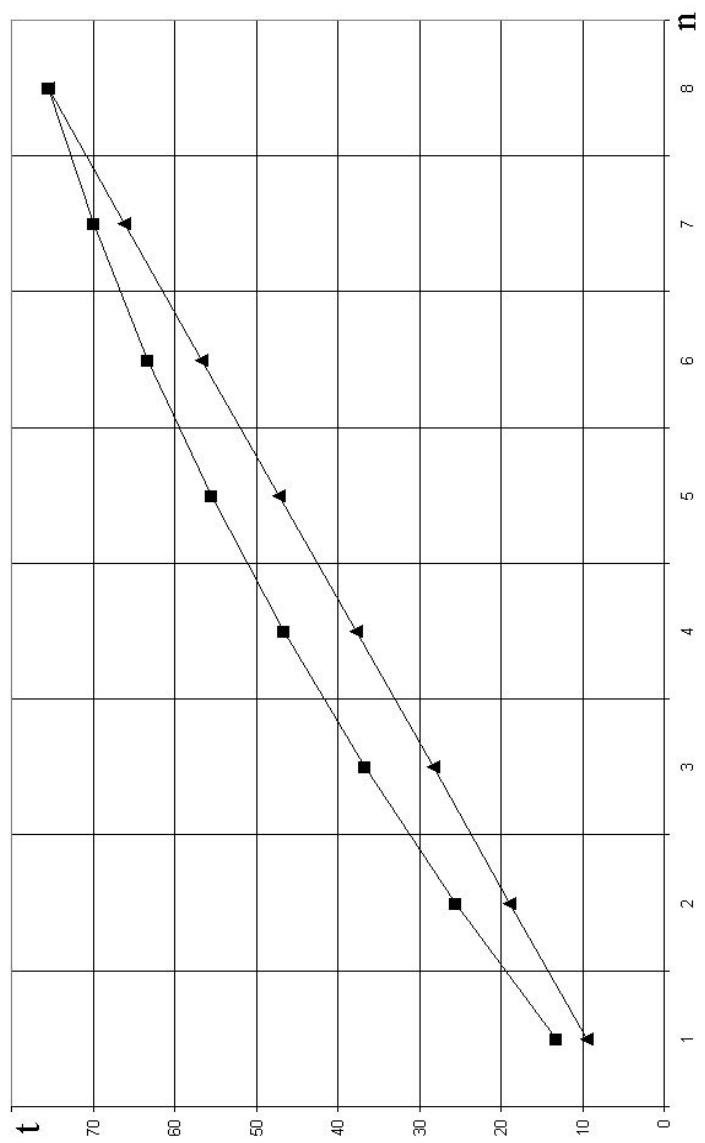


Рис. 12

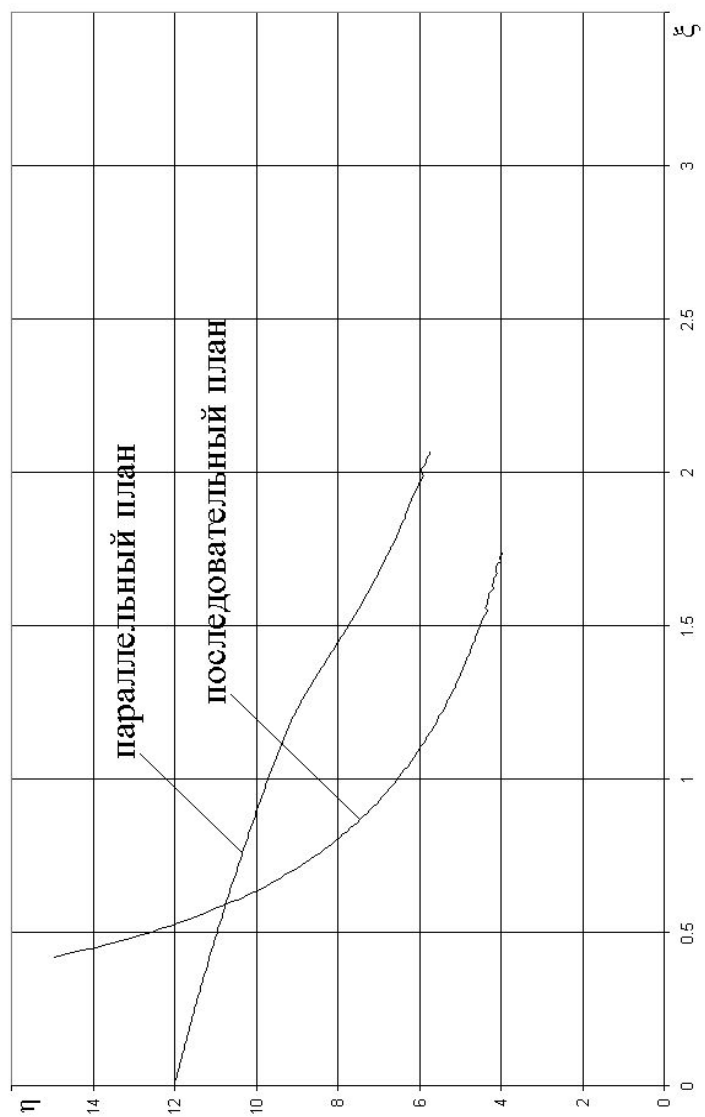


Рис. 13

положительной или отрицательной, усреднялись отдельно положительные и отрицательные значения разности, которые соответственно были обозначены $\Delta\xi^+$, либо с $\Delta\xi^-$; производился подсчёт соответствующих положительных и отрицательных значений Δ , который учитывался с помощью счётчиков Δn^+ и Δn^- . Полученные результаты усреднялись по всем сгенерированным наборам J .

- Таким образом, для каждого эксперимента вычислялись:
- $\bar{\xi}^*$ — среднее значение показателя абсолютного времени исполнения в точке пересечения линии $\langle\eta, \xi\rangle^{par}$ с кривой $\langle\eta, \xi\rangle_{mid}^{seq}$;
 - $\bar{\eta}^*$ — усреднённое значение коэффициента относительного запаздывания выполнения в точке пересечения линии $\langle\eta, \xi\rangle^{par}$ с кривой $\langle\eta, \xi\rangle_{mid}^{seq}$;
 - среднее значение показателя абсолютного времени исполнения для всех задач, имеющих решение в данном наборе;
 - $\bar{\eta}^{par}$ — среднее значение η при $\xi = \bar{\xi}$ для параллельной схемы;
 - $\bar{\eta}_{mid}^{seq}$ — среднее значение η при $\xi = \bar{\xi}$ для последовательной схемы;
 - $\bar{\Delta\xi}^+$ — среднее значение $\Delta\xi^+$ и Δn^+ — доля таких случаев от общего числа наборов для данного эксперимента;
 - $\bar{\Delta\xi}^-$ и Δn^- — соответствующие значения для отрицательных случаев.

Полученные результаты размещены ниже, в табл. 4.

Следует обратить внимание на то, что от эксперимента 1 к эксперименту 4, т.е. при уменьшении числа задач, имеющих решение в наборе:

- значения $\langle\xi^*, \eta^*\rangle$ меняются очень мало;
- значения $\langle\bar{\xi}, \bar{\eta}^{par}\rangle, \langle\bar{\xi}, \bar{\eta}_{mid}^{seq}\rangle$ приближаются к значениям $\langle\xi^*, \eta^*\rangle$ в точке пересечения соответствующих кривых;
- увеличивается доля случаев, когда $\bar{\xi} \leq \xi^*$.

Последнее означает, что большее число (доля) коротких задач

Табл. 4

Тип пакета, соотношение есть решение : нет решения	1 12 : 0	2 10 : 2	3 9 : 3	4 8 : 4
$\overline{\eta}^{par}$ в точке $\bar{\xi}$ для линии $\langle \eta, \xi \rangle^{par}$ если $\bar{\xi} > \xi^*$	9.14	9.71	10.01	10.32
$\overline{\eta}_{mid}^{seq}$ в точке $\bar{\xi}$ для кривой $\langle \eta, \xi \rangle_{mid}^{seq}$ если $\bar{\xi} > \xi^*$	6.63	7.70	8.17	8.58
Δn^+	99.4 %	93.6 %	86.4%	77.4%
$\Delta \xi^+$, если $\bar{\xi} > \xi^*$	0.346	0.215	0.176	0.15
$\bar{\xi}^*$ в точке пересечения	0.657	0.656	0.654	0.650
$\overline{\eta}^*$ в точке пересечения	10.20	10.23	10.29	10.39
Δn^-	0.6 %	6.4 %	13.6%	22.6%
$\Delta \xi^-$, если $\bar{\xi} \leq \xi^*$	0.096	0.099	0.106	0.117
$\overline{\eta}^{par}$ в точке $\bar{\xi}$ для линии $\langle \eta, \xi \rangle^{par}$, если $\bar{\xi} \leq \xi^*$	7.69	8.60	9.11	9.56
$\overline{\eta}_{mid}^{seq}$ в точке $\bar{\xi}$ для кривой $\langle \eta, \xi \rangle_{mid}^{seq}$ если $\bar{\xi} \leq \xi^*$	8.44	9.56	10.34	11.19

получает неоправданно длительную относительную задержку, если выполняются согласно последовательной стратегии.

Аналогично анализировалась точка пересечения кривой $\langle \eta, \xi \rangle_{mid}^{seq}$ с кривой минимальных времен, которая было описана в конце §3. Напомним, что для каждого фиксированного набора J' строится вариационный ряд $X_1^* \leq X_2^* \leq \dots \leq X_n^*$, т.е. заявки упорядочиваются по невозрастанию абсолютных времен выполнения. Предполагается, что таким образом они упорядочиваются в очереди при последовательном выполнении. Далее для них строится кривая $\langle \eta, \xi \rangle_{min}$, и для набора J' вычисляются значения ξ_*, η_* в точке пересечения кривых $\langle \eta, \xi \rangle_{mid}^{seq}$ и $\langle \eta, \xi \rangle_{min}$. Затем определяются усреднённые значения ξ_*, η_* по всем наборам. Результаты моделирования приведены в табл. 5 и на рис. 14.

Табл. 5

Тип пакета.	1	2	3	4
Существование решения, есть : нет	12 : 0	10 : 2	9 : 3	8 : 4
Среднее значение ξ_* в точке пересечения	1.4099	1.4198	1.4402	1.4588
Среднее значение η_* в точке пересечения	4.8576	4.8280	4.7664	4.6934

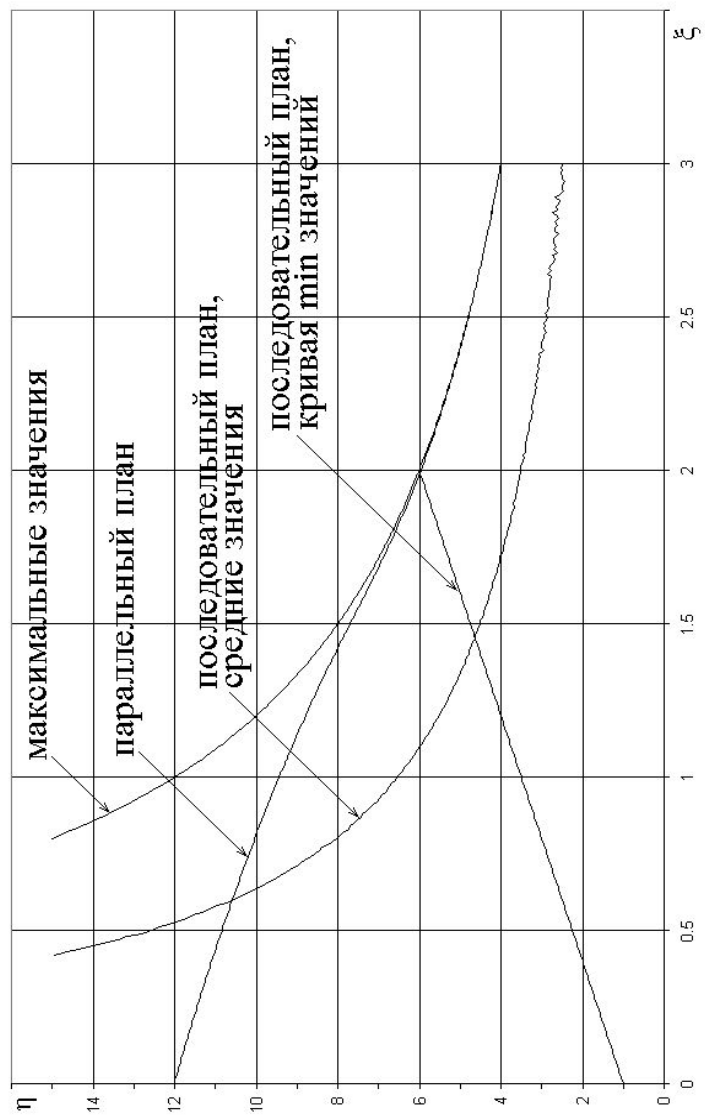


Рис. 14

§5. Обсуждение вероятности существования решения

Рассмотрим процесс решения задачи как процедуру извлечения шаров из урны. При этом будем считать, что в урне либо все шары белые, либо среди шаров есть ровно один чёрный шар. В первом случае будем говорить, что задача "не имеет решения", во втором — что задача "имеет решение". Пусть вероятность того, что все шары в урне белые, равна p , тогда вероятность того, что в урне содержится ровно один чёрный шар, равна $q = 1 - p$.

Предположим, что в начальный момент времени $t = 0$ урна содержит N шаров. Начинаем процесс извлечения шаров из урны в случайном порядке: в момент времени t_1 будет извлечен первый шар, в момент t_2 — второй, и так далее. Процедуру будем продолжать до тех пор, пока не произойдет одно из двух событий (исходов):

- 1). В момент τ^* извлечен шар с порядковым номером k^* и он оказался черным;
- 2). В момент T^* извлечен последний N -й шар и он, как и все предыдущие, оказался белым.

Первый исход означает, что задача решена, второй — что задача решения не имеет. Заметим, что для рассмотренных в предыдущих параграфах задач, $\tau^* = \mathbf{X}$, $T^* = \mathbf{T}$

Рассмотрим промежуточный k -й шаг (момент времени t_k). Пусть из урны извлечено k шаров, и все они оказались белыми. Это означает, что в момент t_k задача не решена, а в урне осталось $N - k$ шаров.

Обозначим через $q(k, N)$ — условную вероятность того, что среди оставшихся $N - k$ шаров есть чёрный, т.е. вероятность того, что решение будет найдено, если продолжить вынимать шары.

Воспользуемся формулой Байеса для апостериорной веро-

ятности гипотезы:

$$P(H_1|A) = \frac{P(H_1) \cdot P(A|H_1)}{\sum (P(H_i) \cdot P(A|H_i))}$$

Пусть гипотеза H_1 состоит в том, что в урне есть чёрный шар. Априорная вероятность этой гипотезы q . Тогда гипотеза H_2 состоит в том, что в урне нет чёрного шара, априорная вероятность для H_2 равна $p = 1 - q$. Событие A состоит в том, что мы извлекли k шаров, и все они белые. Условные вероятности для события A при гипотезах H_1 и H_2 : $P(A|H_1) = (N - k)/N$, $P(A|H_2) = 1$. Тогда по формуле Байеса:

$$\begin{aligned} q(k, N) = P(H_1|A) &= \frac{q \cdot (N - k)/N}{q \cdot (N - k)/N + 1 \cdot p} = \\ &= \frac{1 - p}{1 - p + p \cdot N/(N - k)} = \frac{1 - p}{1 + p/(N/k - 1)}. \end{aligned} \quad (1)$$

или

$$q(k, N) = \frac{N - k}{N/q - k} \quad (2)$$

Обозначим через $p(k, N)$ условную вероятность того, что после извлечения k белых шаров, все оставшиеся шары — $N - k$ — белые, то есть задача не имеет решения. Тогда

$$p(k, N) = \frac{p \cdot N/(N - k)}{1 + p \cdot k/(N - k)}.$$

Обозначим отношение $\frac{q(k, N)}{p(k, N)}$ через $\alpha(k, N)$. Тогда

$$\begin{aligned} \alpha(k, N) &= \frac{q(k, N)}{p(k, N)} = \frac{1 - p}{p \cdot N/(N - k)} = \frac{(1 - p) \cdot (1 - k/N)}{p} = \\ &= \frac{(1 - p) \cdot (1 - \beta(k, N))}{p} = \alpha(0, N) \cdot (1 - \beta(k, N)), \end{aligned}$$

где $\beta(k, N) = k/N$ — доля извлеченных шаров к моменту t_k .

Рассмотрим пример. Пусть $p = 0.25$, тогда $q = 1 - p = 0.75$. $\alpha(0, N) = 0.75/0.25 = 3$, то есть, "априорные шансы" найти решение "да" : "нет" — 3 : 1. При $\beta = k/N = 0.9$, $\alpha(k, N) = \alpha(0.9N, N) = 0.3$, то есть, в момент $t_{0.9N}$ шансы найти решение "да" : "нет" — 3 : 10, что намного меньше, чем в начале.

$$q(0.9N, N) = 3/13 = 0.23, \quad p(0.9N, N) = 10/13 = 0.77.$$

Если сравнить с начальной ситуацией, когда $q(0, N) = 0.75$ и $p(0, N) = 0.25$, то видно, что ситуация изменилась на противоположную.

При тех же условиях — из урны извлекли k шаров и все они белые, рассмотрим другой вопрос: какова вероятность, что $(k + 1)$ -й шар окажется чёрным? Эта вероятность показывает, каковы шансы решить задачу на ближайшем шаге, и её можно использовать для сравнения нескольких вариантов действий. Например, для оценки того, из какой урны "выгоднее" извлекать шар, если урн несколько. Обозначим эту вероятность $p(1, k, N)$ и вычислим её:

$$p(1, k, N) = \frac{1}{N/q - k}.$$

Это выражение можно получить из формулы Байеса. Из формулы видно, что вероятность монотонно возрастает с ростом k . Отсюда следует, что вероятность решить задачу на следующем шаге, если она ещё не решена, больше чем на предыдущем.

При $p = 0.25, q = 0.75$:

$$p(1, 0, N) = \frac{q}{N} = \frac{0.75}{N},$$

$$p(1, 0.9N, N) = \frac{1}{N/0.75 - 0.9N} = \frac{30}{13 \cdot N} = \frac{2.31}{N},$$

$$p(1, N-1, N) = \frac{q}{Np} = \frac{3}{N}.$$

Таким образом, вероятность извлечь чёрный шар (решить задачу), если он еще не найден, на последнем шаге в четыре раза больше, чем на первом. Если вместо шаров оперировать временем, то вероятность решить задачу, если она еще не решена, за последнюю секунду в четыре раза больше, чем за первую.

Рассмотрим ещё один близкий по смыслу вопрос. Пусть имеется две одинаковых урны, в каждой из которых N шаров, и изначальная вероятность того, что в каждой урне есть чёрный шар $q = 0.75$. Нужно как можно быстрее извлечь из урны хотя бы один чёрный шар.

Допустим, что из первой урны последовательно извлекли $0.9N$ шаров, которые оказались белыми, при этом из второй урны шары ещё не доставали, их в ней по-прежнему N . Для того, чтобы как можно быстрее извлечь из урны хотя бы один чёрный шар, в этот момент нужно принять одно из двух решений: вынуть оставшиеся $0.1N$ шаров из 1-й урны или $0.1N$ шаров из 2-й урны. При этом затраченная работа и время будут одинаковы. Определим, в каком случае вероятность найти чёрный шар больше? Если продолжить извлекать шары из 1-й урны, то вероятность извлечь чёрный шар равна вероятности того, что он там есть. Это значение было вычислено ранее, $q(0.9N, N) = 3/13 = 0.23$. При извлечении $0.1N$ шаров из 2-й урны вероятность найти чёрный шар равна $q \cdot 0.1 = 0.75 \cdot 0.1 = 0.075$. Последняя вероятность вдвое меньше, чем в первом случае. Для того, чтобы получить такую же вероятность, какая получена на данный момент для первой урны, нужно извлечь из 2-й примерно $0.3N$ шаров.

Таким образом, когда есть выбор: продолжать решать задачу, в которой уже часть вариантов просмотрена, или начать решать новую задачу (с такими же N и p), то с точки зрения

вероятности нахождения решения в ближайшее время оказывается "выгоднее" продолжать решать "старую" задачу.

Аналогичные соотношения можно выразить через время выполнения. Пусть для переборной задачи обработка каждого варианта требует одно и то же время. Вероятность найти решение в любом варианте одинаковая. Время полного перебора всех вариантов - T . Как и выше предположим, что вероятность найти решение задачи при полном переборе всех вариантов равна q . Пусть с начала решения задачи прошло время t , и решение не найдено.

Введем понятие "интенсивности" получения решений $r(t)$ как математическое ожидание количества полученных решений за единицу времени при условии, что решение еще не найдено. В начале решения задачи интенсивность получения решений равна q/T . Это следует из того, что за малый промежуток времени dt вероятность получить одно решение равна $q \cdot dt/T$ и вероятность получить 0 решений равна $1 - q \cdot dt/T$. Математическое ожидание количества полученных решений за время dt равно $q \cdot dt/T$.

Интенсивность получения решений:

$$r(0) = (q \cdot dt/T)/dt = q/T.$$

В момент времени t интенсивность получения решений при условии, что решение еще не найдено, равна

$$r(t) = \frac{1}{T/q - t} = \frac{q}{T - qt}.$$

Отсюда следует, что если решение ещё не найдено, то интенсивность получения решений со временем возрастает, и выполнение задачи, которая уже некоторое время обрабатывается, надо продолжать.

Л и т е р а т у р а

1. Феллер В. Введение в теорию вероятностей и её приложения. М.: Мир, 1984. Т. 1.
2. Каляев И.А., Левин И.И. Реконфигурируемые мультиконвейерные вычислительные структуры для решения потоковых задач обработки информации и управления // Труды международной научно-практической конференции "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный университет, 2010. Т. 1. С.100-102.
3. Забродин А.В., Левин В.К., Корнеев В.В. Массово-параллельные системы МВС-100 и МВС-1000. // Сборник научных трудов научной сессии МИФИ - 200. М.: МИФИ, 2000. Т.2. С. 194-195.
4. Баранов А.В., Киселев А.В., Корнеев В.В. и др. Управление сетевой средой распределенных вычислений // Труды Первой Всероссийской научной конференции "Методы и средства обработки информации". М.: МГУ им. М.В. Ломоносова, 2003. С. 98-103.
5. Баранов А.В., Киселев А.В., Корнеев В.В. и др. Программный комплекс "Пирамида" организации параллельных вычислений с распараллеливанием по данным // Труды международной суперкомпьютерной конференции и конференции молодых ученых "Научный сервис в сети Интернет: Суперкомпьютерные центры и задачи". М.: МГУ им. М.В. Ломоносова, 2010. С. 299-302.
6. Garonne V., Stokes-Rees I., Tsaregorodsev A. DIRAC: A Sca-

- lable Lightweight Architecture for High Throughput Computing // Proc. 5th IEEE/ACM Int. Worksh. on Grid Computing. IEEE Comp. Sc.: Washington, 2004. P. 19 - 25.
7. Garonne V., Tsaregorodtsev A., Caron E. A study of meta-scheduling architectures for high throughput computing: Pull versus Push // Proc. 4th Int. Symp. on Parallel and Distribut. Computing. IEEE Comp. Sc.: Washington, 2005. P. 226-233.
 8. Dongarra J., Foster I., etc. edit. Sourcebook of parallel computing. San Francisco: Morgan Kaufmann Publ. Inc., 2003.
 9. Casavant T., Kuhl J. Taxonomy of Scheduling in General Purpose Distributed Computing Systems // IEEE Transactions on Software Engineering. 1988. V. 14. N. 2. P. 141-154.
 10. Matarneh R. Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes // Am. J. Applied Sci., 2009. V. 10. N. 6. P. 1831-1837.
 11. Ронжин А.В., Суриков В.Н. О математических проблемах применения высокопроизводительных вычислительных систем для решения задач случайного поиска // Труды международной научно-технической конференции "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный университет, 2010. Т. 2. С. 239-243.
 12. Голосов П.С., Козлов М.В., Малащенко Ю.Е. и др. Модель системы управления специализированным вычислительным комплексом. М.: ВЦ РАН, 2010.

13. Adams D. The Hitchhiker's Guide to the Galaxy. New York:
Del Rey, 1995.

СОДЕРЖАНИЕ

Введение	3
§1. Общее описание моделей	7
§2. Последовательное выполнение заданий	12
§3. Параллельное выполнение заданий	21
§4. Вычислительный эксперимент	30
§5. Обсуждение вероятности существования решения	52
Литература	57

Список литературы

- [1] Феллер В. Введение в теорию вероятностей и ее приложения. М.: Мир, 1984. Т. 1.
- [2] Каляев И.А., Левин И.И. Реконфигурируемые мультиконвейерные вычислительные структуры для решения потоковых задач обработки информации и управления // Труды международной научно-практической конференции "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный университет, 2010. Т. 1. С.100-102.
- [3] Забродин А.В., Левин В.К., Корнеев В.В. Массово-параллельные системы МВС-100 и МВС-1000. // Сборник научных трудов научной сессии МИФИ - 200. М.: МИФИ, 2000. Т. 2. С. 194-195.
- [4] Баранов А.В., Киселев А.В., Корнеев В.В. и др. Управление сетевой средой распределенных вычислений // Труды Первой Всероссийской научной конференции "Методы и средства обработки информации". М.: МГУ им. М.В. Ломоносова, 2003. С. 98-103.
- [5] Баранов А.В., Киселев А.В., Корнеев В.В. и др. Программный комплекс "Пирамида" организации параллельных вычислений с распараллеливанием по данным // Труды международной суперкомпьютерной конференции и конференции молодых ученых "Научный сервис в сети Интернет: Суперкомпьютерные центры и задачи". М.: МГУ им. М.В. Ломоносова, 2010. С. 299-302.
- [6] Garonne V., Stokes-Rees I., Tsaregorodsev A. DIRAC: A Scalable Lightweight Architecture for High Throughput

- Computing // Proc. 5th IEEE/ACM Int. Workshop on Grid Computing. IEEE Computer Society, Washington, 2004. P. 19 - 25.
- [7] Garonne V., Tsaregorodtsev A., Caron E. A study of meta-scheduling architectures for high throughput computing: Pull versus Push // Proc. 4th Int. Symp. on Parallel and Distributed Computing. IEEE Computer Society, Washington, 2005. P. 226-233.
 - [8] Dongarra J., Foster I., etc. edit. Sourcebook of parallel computing. San Francisco: Morgan Kaufmann Publishers Inc., 2003.
 - [9] Casavant T., Kuhl J. Taxonomy of Scheduling in General Purpose Distributed Computing Systems // IEEE Transactions on Software Engineering. 1988. V. 14. N. 2. P. 141-154.
 - [10] Matarneh R. Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes // Am. J. Applied Sci., 2009. V. 10. N. 6. P. 1831-1837.
 - [11] Ронжин А.В., Суриков В.Н. О математических проблемах применения высокопроизводительных вычислительных систем для решения задач случайного поиска // Труды международной научно-технической конференции "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный университет, 2010. Т. 2. С. 239-243.
 - [12] Голосов П.С., Козлов М.В., Малащенко Ю.Е., Назарова И.А., Ронжин А.Ф. Модель системы управления специа-

лизированным вычислительным комплексом. М.: ВЦ РАН, 2010.

- [13] Adams D. The Hitchhiker's Guide to the Galaxy. New York: Del Rey, 1995.