

Наной

68

О СТАТИСТИЧЕСКИХ ЗАДАЧАХ НА ВЫЧИСЛИТЕЛЬНЫХ

СИСТЕМАХ

М. АРАТО - Будапешт

1979. г. апрель

О СТАТИСТИЧЕСКИХ ЗАДАЧАХ НА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

М. АРАТÓ - Будапешт

1. Для статистических измерений и выводов в настоящее время всё больше и больше используются электронные вычислительные машины.

В венгерской Народной Республике мы различаем 4 типа задач, где происходит машинная обработка данных и статистические методы оказываются нужными и полезными.

1. Измерение и управление процессов,
2. Управленческие и экономические процессы на предприятиях и заводах,
3. Государственная статистика,
4. Научные и технические расчеты.

В докладе подробно изучаются те методы и средства, которые дают возможность эффективно решать поставленные задачи.

В задачах измерения и управления процессами используются в основном, малые машины с возможностями работы в реальном масштабе времени. Математическим аппаратом является статистика случайных процессов или временных рядов /стохастические дифференциальные уравнения/. Время обработки и ответа измеряется в секундах.

На предприятиях статистическая обработка ведется в основном в таких задачах как управления с ресурсами и в контроле производства. Математические модели в основном берутся из теории надежности, массового обслуживания и многомерного анализа. Используются специальные пакеты программ и на больших машинах программные системы для баз данных.

Время обработки и ответа измеряется в часах и днях.

В государственной статистике в основном используются большие специфические пакеты программ и программные системы баз данных

/ MARK IV, IDS, IDMS/. Математический аппарат: многомерный анализ и временные ряды.

Время обработки и ответа измеряется в месяцах.

В научных расчетах в основном используется пакет программ как BMDP. Математический аппарат специфический. Время обработки и ответа является неопределенным.

2. ПОВЕДЕНИЕ ПРОГРАММ И МУЛЬТИПРОГРАММИРОВАНИЕ

2.1. Контроль и распределение ресурсов памяти в виртуальной системе с мультипрограммным режимом является одним из главных задач вычислительной математики. Виртуальная память может определяться как оперативная память имитационной вычислительной машины. Содержание информации отдельных программ распределено на блоки одинакового размера, которые называются страницами /page/. Результаты управления с памятью формируются в терминах страниц. Так называемая последовательность референций $\xi_1, \xi_2, \dots$ /reference string/ определяется следующим образом $\xi_t = i$ если в момент t в процессе работы потребовалось содержание i -й страницы / $i=1, 2, \dots, n$ и $t=1, 2, \dots$ /. Эта последовательность используется для описания поведения программ.

Уже упомянутая рабочая часть программы обозначает то множество страниц, которое должно быть в оперативной памяти, чтобы обеспечить определенный уровень для эффективной работы программ. Практическим определением рабочей части программ занимались многие авторы, но все попытки имеют описательный характер пока не дадим строго стохастическую характеристику последовательности ξ_t .

Основные типы поведения программ, которые встречаются на практике следующие.

М. АРАТО

- а/ ξ_t является независимой последовательностью с одинаковым распределением /independent reference string/: $P\{\xi_t=i\}=p_i$
- в/ ξ_t образует цепь Маркова с рекуррентными состояниями и свойством перемешивания. Матрица переходов $\{p_{ij}\}=\{P(\xi_{t+1}=j|\xi_t=i)\}$ не зависит от t .
- с/ LRU стэк модель, где в оперативной памяти на первом месте стоит та страница, которая вызывалась последний раз. При вызове i -ой страницы /в оперативной памяти помещается $m < n$ страниц/, если $i < m$, страница перемещается на первое место и 1-ое по $(i-1)$ -й странице сдвинутся на одно место ниже, остальные страницы останутся на месте, если $i > m$ страница со вспомогательной памяти /со второго уровня/ берется на 1-ое и m -ая страница выбрасывается на второй уровень памяти, остальные страницы на одно место ниже сдвинутся. Взятие страницы со второго уровня называется страничной ошибкой /page fault/. Предполагается, что обращение к страницам следующее: вероятность вызова с i -ого места равна q_i и последовательные вызовы являются независимыми.

Последний тип поведения программ является обобщением следующей схемы: i -я страница в момент t имеет вероятность вызова $a_i (>0)$, а потом эта страница вызывается v_i раз, v_i имеет геометрическое распределение

$$P\{v_i=k\} = (1-b_i)b_i^{k-1}, \quad (0 \leq b_i \leq 1)$$

Статистические анализы показывали, что ни один из упомянутых моделей не описывает, с достаточной точностью, поведение программ.

Вероятностные модели в вычислительных системах

Рассматривается следующая задача автоматического управления в операционной системе, когда предполагается, что два принципа поведения программ чередуются в случайные моменты времени. Предположим, что с моделью поведения а/ система работает случайное время ξ_2 , потом работает в режиме с/ случайное время ξ_2 снова в режиме а/ в длительности ξ_3 .

Пусть π обозначает вероятность события, что в начальный момент система работает в режиме а/. Если обозначим случайный процесс

$$\eta(t) = \begin{cases} 1 & \text{в момент } t \text{ работает режим а/} \\ 0 & \text{в момент } t \text{ работает режим с/} \end{cases}$$

то он не подлежит наблюдению, наблюдается только процесс референций $\xi_1, \xi_2, \dots$. По предложению

$$P\{\eta(0) = 1\} = \pi,$$

и кроме того, ради упрощения, предположим, что величины ξ_i являются независимыми случайными величинами с одинаковым распределением

$$P\{\xi_i = k\} = p^{k-1}(1-p),$$

где $0 \leq p \leq 1$ известная константа.

На основании наблюдений $\xi_1, \xi_2, \dots$ надо найти такую последовательность Марковских моментов $\tilde{\xi}_1, \tilde{\xi}_2, \dots$ что если t фиксированный момент времени и

М. АРАТЌ

$$N_t = \{ \max n: \tilde{\xi}_1 + \dots + \tilde{\xi}_n < t \},$$

то при условии

$$E N_t \leq t \frac{c}{1-p},$$

где с фиксированная константа, минимизируется математическое ожидание

$$E \left(\int_0^t \chi_s ds \right).$$

Здесь

$$\chi_s = \begin{cases} 1 & \text{при } s \leq \tilde{\xi}_1, \tilde{\xi}_1 \leq s \leq \tilde{\xi}_1 + \tilde{\xi}_2, \tilde{\xi}_1 + \tilde{\xi}_2 \leq s \leq \tilde{\xi}_1 + \tilde{\xi}_2 + \tilde{\xi}_3, \dots \\ 0 & \text{в остальных случаях} \end{cases}$$

т.е. $\chi_s = 1$ если решение по наблюдениям относительно режима работы не соответствует истинному режиму.

Замечание 1. В этой задаче нельзя взять ту же функцию риска, как это сделали в пункте 2.1, разладки, так как в той задаче предполагается, что имеется возможность при обнаружении разладки проверять произошла ли она или нет. В настоящей задаче такой возможности нет и поэтому нельзя отдельно минимизировать $E(\tilde{\xi}_1 - 0)$, $E(\tilde{\xi}_1 + \tilde{\xi}_2 - \tilde{\xi}_1 | \tilde{\xi}_1 + \tilde{\xi}_2 > \xi_1)$

Точное решение этой задачи неизвестно. В том случае, когда

$\frac{1}{1-p} \gg 1$, т.е. система долго находится в одном режиме применяем метод последовательного анализа для различения

двух гипотез следующим образом. Пусть P_a , соответственно P_e , обозначают распределения при режиме а/, соответственно с/.

Выбираются значения $b_1 < b_2$ таким образом, чтобы

$$b_1 < \log \frac{P_a(\xi_1, \dots, \xi_n)}{P_c(\xi_1, \dots, \xi_n)} < b_2$$

и первый выход ν за пределами $[b_1, b_2]$ имел математические ожидания

$$E_a \nu = \frac{1}{n_0} \frac{1}{1-p}, \quad E_b \nu = \frac{1}{n_0} \frac{1}{1-p},$$

где p_0 заданное число (>1). После принятия решения отбрасываются предыдущие наблюдения ξ и снова применяется предыдущая процедура. Если $\nu \gg \frac{1}{1-p}$ то отбрасываются предыдущие

Вероятностные модели в вычислительных системах

наблюдения и процесс последовательного решения заново начинается.

Замечание 2. Неизвестны простые, но достаточно мощные, критерии для различия двух гипотез: независимая модель $a/$, и модель $c/$.

2.2 В этом пункте рассматриваем задачи связанные с замкнутыми линейными сетями, где обслуживающие приборы связаны между собой и время обслуживания не экспоненциальное.

Формулировкам такого типа задач с разными устройствами, как например центральные процессоры, каналы ввода и вывода, пульты пользователей, коммутаторы и т.д., мы уже привыкли. Чтобы показать силу стохастического анализа покажем один пример, где системами обслуживания являются процессы обработки, которые используют разные ресурсы вычислительной системы. Таким образом прибором обслуживания есть программа и предмет нашего изучения математическое обеспечение. В математическом обеспечении произвести изменения намного легче чем в устройствах. Пусть рассматривается маленькая вычислительная машина с несколькими дисплеями, одним центральным процессором, с дисками и магнитными лентами для одновременного ввода данных или программ на магнитные ленты. Такие системы используются вместо известных перфокарточных устройств, чтобы существенно сократить временной интервал между вводом и проверкой данных. Предположим, что фиксируются стандартные, но разного типа форматы /можно представить канцелярские, отчетные документы/. В памяти находятся разные типы программ:

1. супервизоры для управления дисплеями,
2. программы для работы со словарями и разными форматами,
3. программы для работы с файлами,

4. программы для обработки данных.

Все эти программы или их типы служат для данных рекордов обслуживающими приборами и чтобы рекорды зафиксировались на магнитную ленту должны много раз пребывать на этих приборах. После обслуживания на одном приборе рекорд /требование/ с определенными вероятностями может переходить на другой прибор. Чтобы сформулировать задачу, или чтобы определить состояние сети, а это означает числа $n_i(t)$, $i = 1, 2, \dots, M$, требований в приборах в момент t , надо дать число приборов M , число требований N , матрицу перехода (p_{ij}) , интенсивность μ_i ($i = 1, 2, \dots, M$) распределение обслуживания требований приборами. Схема для вышеописанной системы программ иллюстрируется на рисунке 1.

В том случае когда время обслуживания на разных приборах имеет показательный закон хорошо известно /см. например [10]/, что стационарное распределение состояния системы можно получить в форме

$$(*) \quad P(n_1, \dots, n_m) = \frac{\sum_{j=1}^M \left(\frac{p_j}{\mu_j} \right)^{n_j}}{\sum_{(N)} \prod_{l=1}^M \left(\frac{p_l}{\mu_l} \right)^{n_l}}, \quad \sum_{j=1}^M n_j = N,$$

где p_i является решением системы

$$p_i = \sum_{j=1}^M p_{ji} p_j, \quad i = 1, 2, \dots, M,$$

при нормирующем условии $\sum_i p_i = 1$. Эти формулы показывают, что система в пределе ведет себя как совокупность M независимых приборов.

Из многочисленных приближений для общего случая когда распределения не являются показательными здесь напомним два.

Вероятностные модели в вычислительных системах

Первое приближение для стационарного режима использует результаты экспоненциальных распределений в следующей форме. Если в формуле (*) нормирующий множитель имеет форму $(1 - p_i/\mu_i)$ для i -ого прибора, т.е.

$$p_i(n_i) = (1 - p_i/\mu_i) \left(\frac{p_i}{\mu_i}\right)^{n_i},$$

то прибор может рассматриваться как модель $M/M/1$. В том случае когда нормирующим множителем выбирается

$$1 / \sum_{k=0}^N (p_i/\mu_i)^k, \quad \text{т.е. для } j\text{-ого прибора}$$

$$p_i(n_i) = (p_i/\mu_i)^{n_i} / \sum_{k=0}^N (p_i/\mu_i)^k$$

то прибор можно рассматривать как $M/M/1/N$ модель, где интенсивность входного потока p_i , интенсивность обслуживания μ_i , и N является размером длины очереди.

В общем случае предполагается, что стационарный режим также распадается на M отдельных компонентов, где i -й прибор можно рассматривать как $M/G/1$ или $M/G/1/N$ (см[13]).

Вторым методом является приближение чисел требований в отдельных приборах винеровским процессом /см. [10], [11]/. Этот подход использовался только для систем с двумя приборами и N требований. Пусть обозначают μ_1, μ_2 интенсивность обслуживания, и коэффициенты вариаций

$$k_1 = \sigma^2/(\mu_1^2), \quad k_2 = \sigma^2/(\mu_2^2).$$

Введем величины

$$a = \mu_1 - \mu_2, \quad b = \mu_1 k_1 + \mu_2 k_2.$$

/сность и коэффициент диффузии/, тогда для числа требования k у первого прибора получается $(x=k/N)$, что в стационарном режиме вероятность пустой системы равна $p=1 - \frac{\mu_1}{\mu_2}$, а плотность вероятностей для x имеет вид

М. АРАТÓ

$$\frac{\mu_1}{\mu_2} [1 - e^{\gamma x}], \text{ где } \gamma = \frac{2\alpha}{b}.$$

Было бы желательно разработать диффузионное приближение для схем изображенных на рисунке 2.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

- [1] АРАТÓ М. Diffusion approximation for multiprogrammed computer systems.
Comp. and Maths. with Appls. Vol. 1. (1975) 315-326.
- [2] АРАТÓ М. Statistical sequential methods for utilization in performance analysis.
Modelling and Performance Evaluation of Computer Systems. H. BEILNER and E. GELENBE (eds.), (1977), North-Holland. 287-304.
- [3] АРАТÓ М. - KNUTH E. - TÖKE P. On stochastic control of a multiprogrammed computer based on a probabilistic model.
IFAC Symp. on Stochastic Control, Budapest, (1974)
- [4] BEILNER H. - GELENBE E. Modelling and Performance Evaluation of Computer Systems, (1977), North-Holland.

- [5] BENCZÜR A. - KRÁMLI A. A note on data base integrity. Acta Cybernetica, Szeged, (1977), 181-185.
- [6] CHANDY K. - REISER M. Computer Performance (1977), North-Holland.
- [7] CHANDY K. - BROWNE J.C. - DISSLY C.W. - UHRIG W.R. Analytical models for rollback and recovery strategies in data base systems. IEEE Trans. Software Eng. 1(1975) N°1, 100-110.
- [8] DAVIS M.H.A. A note on the Poisson disorder problem, Banach Center Publications, Vol. 1, 65-72, (1976).
- [9] ГАЛЬЧУК Л. - РОЗОВСКИЙ Б. Разладка Пуассоновского процесса, Теория Вер. и ее Прим. /1971/, 729-734.
- [10] GELENBE E. - MUNTZ R. Probabilistic models of Computer Systems, Acta Informatica 7 (1976) 35-60.
- [11] JAISWAL N.K. Priority Queues, Academic Press, London (1968) (на русском языке Очереди с приоритетами, МИР, 1973).
- [12] ШИРЯЕВ А.Н. Статистический последовательный анализ Наука, /1969/, Москва.
- [13] SHUM A. - BUZEN J. A method for obtaining approximate solutions to closed queueing networks with general service times. Modelling and Performance evaluation H. BEILNER and E. GELENBE (eds.) (1977), North-Holland, 201-220.

M. APATÒ

- [14] SEGALL A. Dynamic file assignment in a computer network.
IEEE Trans. on Automatic Control, Vol. AC-21, (1976).
- [15] TOMKÓ J. Processor utilization study.
Comp. and Maths. with Appls. 1 (1975), 337-344.

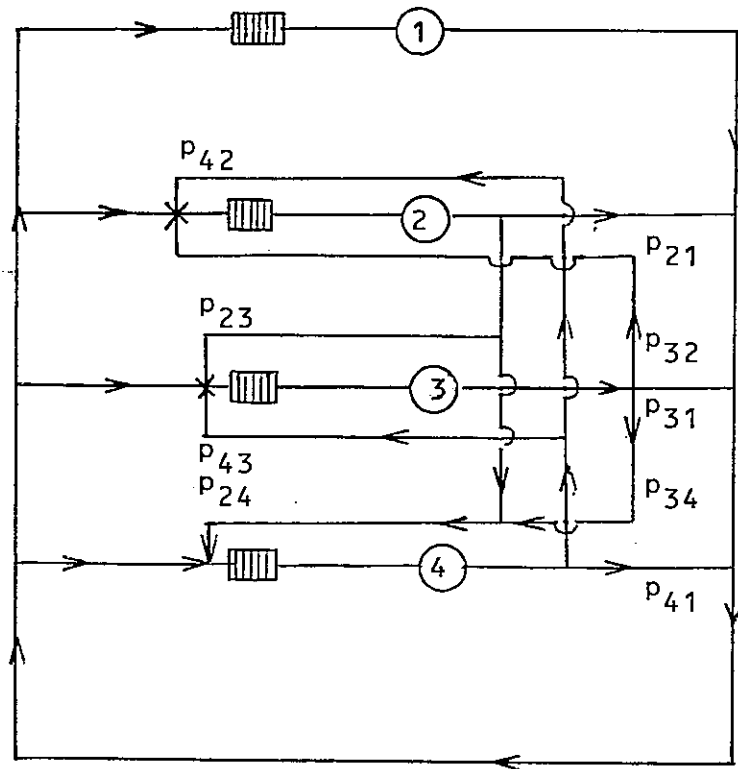


Рис. 1

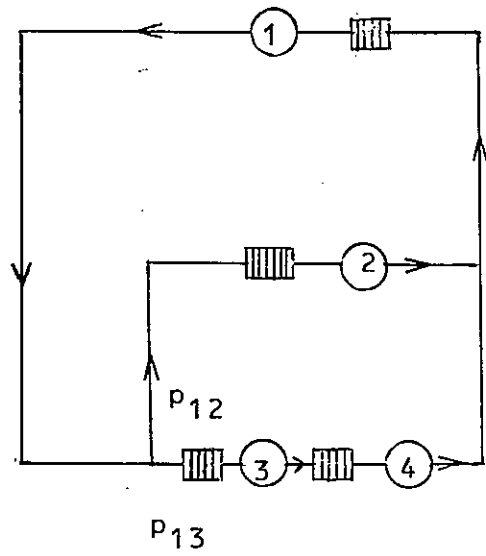
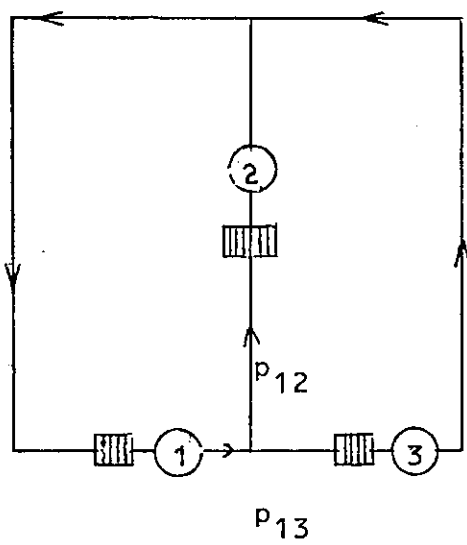


Рис. 2

3. ON PLACEMENT OF RECORDS IN LINEAR STORAGE

In a recent paper McKellar and Wong [3] raised the problem to allocate a set of records $R_1, R_2, \dots, R_n$ in a linear storage in such a way that the expected access time is minimized when the access probabilities are not known. When the consecutive accesses $\xi_1, \xi_2, \dots, \xi_t, \dots$ ($\xi_t = i$ when at time moment t the i -th record was accessed) are independent, identically distributed, with probability distribution

$$P\{\xi_t = i\} = p_i$$

and $\{p_i\}$ ($i = 1, 2, \dots, n$) are known the optimal solution to minimize the expected distance between consecutive request

$$E(D) = \sum_{i,j=1}^n p_i p_j d(i,j)$$

(where $d(i,j)$ denotes the distance between records R_i and R_j) is well known. It is the so called organ pipe arrangement. If we assume, that for simplicity $p_1 \leq p_2 \leq \dots \leq p_n$,

the optimal arrangement is on Figure 1.

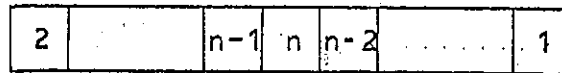


Fig. 1

We make now the following two remarks (using the Euclidean distance):

- a) For uniform distribution ($p_j = \frac{1}{n}$; $j = 1, 2, \dots, n$), or for random placement of the records,

$$E_u(D) \sim \frac{n}{3} \quad (1)$$

- b) When $p_j = \frac{2j}{n(n+1)}$, the linear growing case, and in optimal arrangement

$$E_L(D) \sim \frac{7n}{30}, \quad (2)$$

(see e.g. formula (16) in [3]). In both cases $n \rightarrow \infty$.

Let $f_i(t)$ denote the frequency of access of record R_i until time t . Then

$$\sum_{i=1}^n f_i(t) = t, \quad (3)$$

and if we are waiting for all accesses of the n records

$$\tau = \sum_{i=1}^n f_i(\tau) ; \quad (4)$$

where $f_i(\tau) \geq 1, i = 1, 2, \dots, n$.

τ is the first time (random variable) when all records were accessed. From the law of large numbers

$$\frac{f_i(t)}{\sum_{i=1}^n f_i(t)} = \frac{f_i(t)}{t} \sim p_i \quad (5)$$

and

$$\sum_{i,j=1}^n \frac{f_i(t)f_j(t)}{t^2} d(i,j) \sim \sum_{i,j=1}^n p_i p_j d(i,j) \quad (6)$$

For the random variable τ the same is true (see e.g. R  NYI [4])

$$\frac{f_i(\tau)}{\tau} \sim p_i \quad (5')$$

and

$$\begin{aligned} \sum_{i,j} \frac{f_i(\tau)f_j(\tau)}{\tau^2} d(i,j) &= \\ &= \sum_{i,j} \frac{f_i(\tau)f_j(\tau)}{(\sum_i f_i(\tau))^2} d(i,j) \sim \sum_{i,j} p_i p_j d(i,j). \end{aligned} \quad (6')$$

From the above assumptions we see that the random variables $f_i(t), i = 1, 2, \dots, n$, are polinomially distributed. (They cannot be uniformly and independently distributed as it was assumed in [3]   3. Later in [3]   4-7. it is assumed that $f_i(t)$ are polinomially

distributed, and further that the original distribution is $p_i = \frac{1}{n}$, in which case does not exist the problem, because our formula (1) is true.)

First of all we recall the following statement, which the reader may find in book [2] (Ch. I, § 2, theorem 1, p. 18).

Lemma 1. When $p_i = \frac{1}{n}$ ($i = 1, 2, \dots, n$) and $n \rightarrow \infty$, the expectation

$$E(\tau) = E\left(\sum_{i=1}^n f_i(\tau)\right) = n \sum_{i=1}^n \frac{1}{i} \sim n \log n, \quad (7)$$

and variance

$$D^2(\tau) = n^2 \sum_{i=1}^n \frac{1}{i^2} - E\tau \sim \frac{\pi^2}{6} n^2 - n \log n. \quad (8)$$

Remark 1. From (7) and (8) we get that

$$E\left(\sum_{i=1}^n f_i(\tau)\right)^2 = E(\tau)^2 \sim \left[\frac{\pi^2}{6} + \log^2 n\right] n^2 \quad (9)$$

$\sim n \log n,$

which contradicts formula (12) in [3], where the authors got the wrong

$$E\left(\sum_{i=1}^n f_i\right)^2 \sim \frac{n^2}{4} + \frac{n}{12}$$

asymptotic result. (Further it has to be noticed that in their basic lemma (54) the authors in [3] take expectation

in the following form

$$E\left(\frac{\xi}{\eta}\right) \sim \frac{E(\xi)}{E(\eta)}$$

and this is one of the reasons that the lemma cannot be valid.)

Remark 2. As the basic lemma in [3] is wrong theorems 2. and 3. are also wrong. That the "theorems" are not true can be easily seen from the following simple consideration. The average value of distance is distribution dependent when $\{p_i\}$ is known, it remains distribution dependent in case, when $\{p_i\}$ is not known. For uniform distribution (or approximately uniform distribution) it is $\frac{n}{3}$ and cannot be improved! Of course when the number of accesses to records is $n \log n$ the law of large numbers works and for distribution in (2) the optimal static placement gives asymptotically $\frac{7n}{30}$.

Remark 3. Lemma 1. shows that time until all the records accessed has expectation $n \log n$. It may be proved (see [2] p.20) that more exactly

$$\lim_{n \rightarrow \infty} P\left\{\frac{\tau}{n} - \log n < x\right\} = e^{-e^{-x}}.$$

Remark 4. Let we have a polynomial distribution $p_i \sim \frac{1}{i^n}$ and $p_1 \leq p_2 \leq \dots \leq p_n$ with $t \cdot p_1 \rightarrow \infty$ ($t \rightarrow \infty$), and $\alpha_1 < \frac{t}{n} - \log n < \alpha_2$, then if $\mu(t)$ is the number of not accessed records

$$P\{\mu(t) = 0\} \sim e^{-\theta} \quad \text{if } n \rightarrow \infty$$

where $(1 - \frac{\alpha}{n})t \sim \left(\frac{\theta}{n}\right)^\alpha$ and $\lim_{n \rightarrow \infty} \sum_{i=1}^n \left(\frac{\theta}{n}\right)^{p_i \cdot n} = \theta$,
 $\left(\frac{\theta}{n}\right)^{p_1 \cdot n} \rightarrow 0$, $p_n \cdot n < c$. (see [1]).

The question of dynamic placement of records remained open.

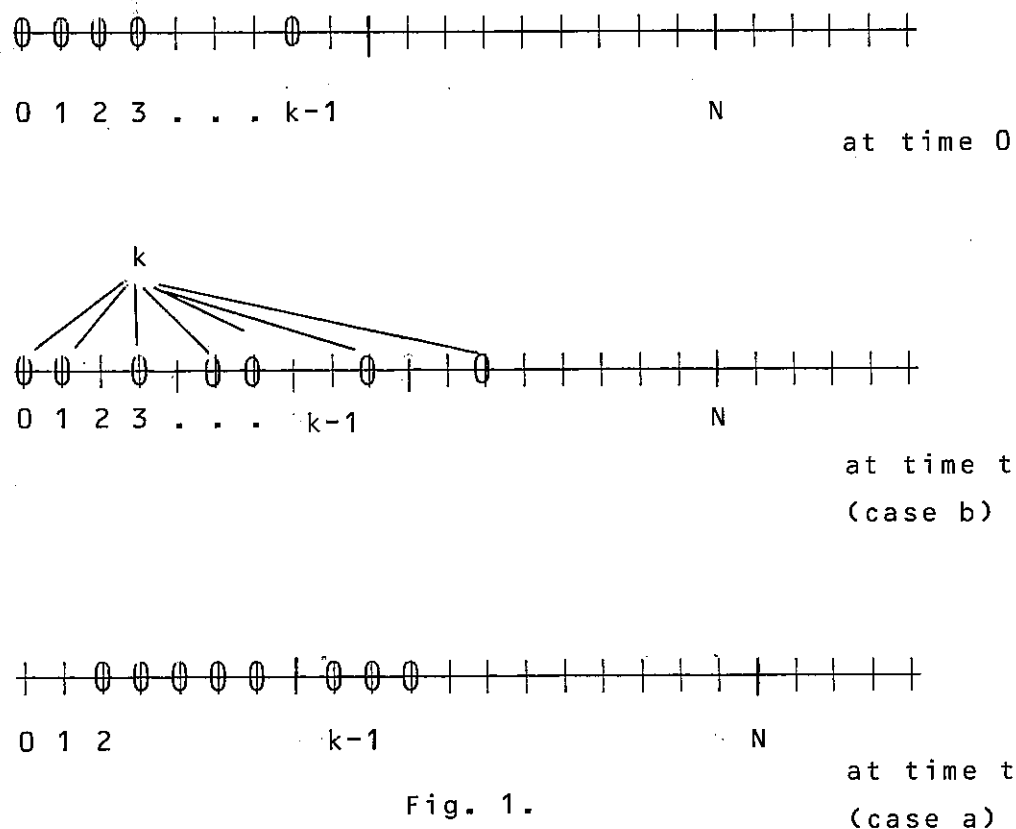
We want to return to this question in a separate paper.

REFERENCES

- [1] BOLOTNIKOV Yu. Limit processes in a model of unequal probabilities arrangement of particles in cells. Theory of Prob. and its Applic. (1968) N° 3, 534-542.
- [2] KOLCHIN, V., SEVASTYANOV, B., CHISTYAKOV, V. Random occupancy problems (in Russian) Nauka, Moscow, (1976).
- [3] McKELLAR, A., WONG, C. Dynamic Placement of Records in Linear Storage Journ. of ACM (1978) N°3, 421-434.
- [4] RÉNYI A. Probability theory (in Hungarian), Akadémiai Kiadó, (1968).

4. ANALYTICAL METHODS IN DISTRIBUTED SYSTEMS

In a packet switched network there are two possibilities to send k packets from node O to node N /on a real line/. In the first case a/ all the packets are moving together, and in the other case b/ the packets are



moving independently /if one can do it/. After t moments /or time/ one may have the situation which is described on Fig.1. For simplicity let us assume that we have k balls /at time moment $t=0$ / on the points $0, 1, \dots, k-1$. Each ball has the possibility to move right with probability $1-p$ and to remain on the same

place with probability p . In the first model /model a// the balls are connected, this means they can move only together, and the probability to make 1 step is

$$q_k = (1-p)^k$$

and after time t to make i steps is

$$\binom{t}{i} q_k^i (1-q_k)^{t-i}.$$

In the second model /model b// a ball, on place j , with probability p , under condition that it is possible or with probability 1, under condition it is impossible.

Let τ denote the time when the first ball /which is at time moment $t=0$ in O / will be in N . What is the expectation:

$$E\tau$$

in case a/ and b/? It is easy to prove that in case a/ we have

$$\begin{aligned} E_a \tau &= \sum_{i=0}^{\infty} (N+1) \binom{N+1}{i} q_k^N (1-q_k^U)^i = N+(N+1) \frac{1-q_k^U}{q_k^U} = \\ &= \frac{N+1}{q_k^U} \end{aligned}$$

On the other side it can be proved that

$$E_b \tau \leq E_a \tau$$

but the exact value of $E_b \tau$ is not known. This exercise is only a simple example for analytical methods in distributed systems. In the following we shall discuss some other problems of distributed systems.

A growing interest exists in distributed systems, and there is little doubt that distributed computing will be the primary systems architecture of the future. Systems are evolving toward a greater diffusion of processing power and databases.

The advantages of decentralised computing can be further enhanced by linking the distributed parts through electronic communications.

There are

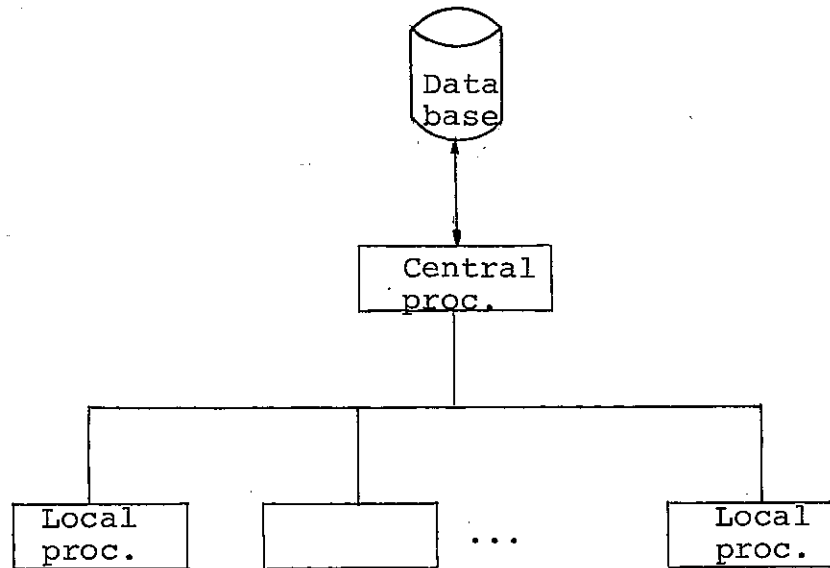
- a/ centralized systems,
- b/ decentralized systems,
- c/ variations on the centralized/decentralized structures.

Distributed systems /main features/:

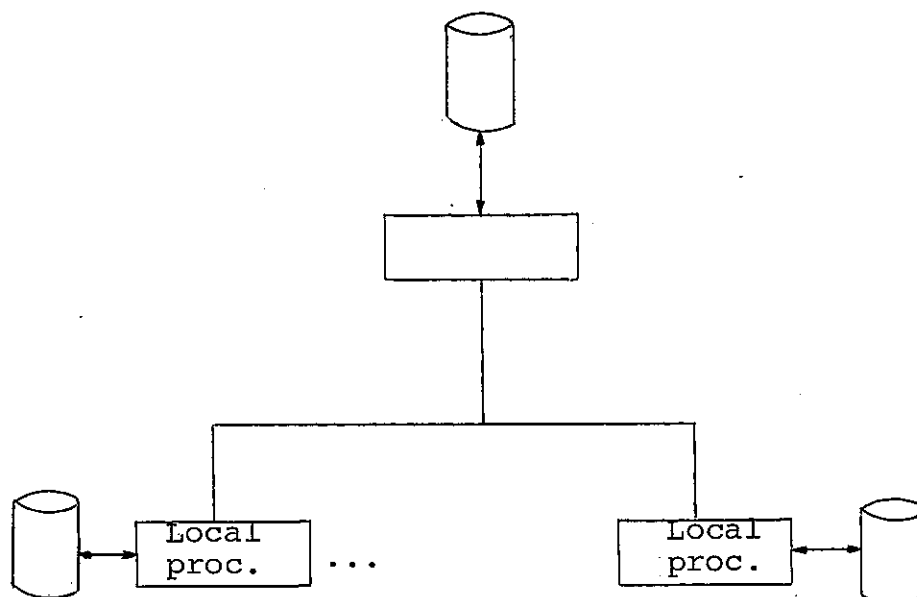
- multiple processors having general purpose computing compatibilities,
- communication links /often only intermittent/ among processors;
- relatively weak interactions among the distributed subsystems;

- considerable centralized coordination in the design and operation of the separate processing subsystems.

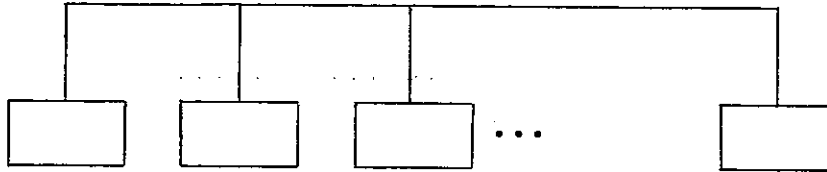
1. Distributed processing without local database



2. Hierarchical systems with non-shared local databases



3. Fully distributed network



CENTRALIZATION OR DECENTRALIZATION

1. Centralization /advantages and disadvantages/

- hardware economies of scale,
- operating economies of scale,
- more powerful capabilities,
- demand smoothing,
- reduced number of facilities,
- development of professional staff,
- increased system integration.

2. Decentralization

- greater control by users,
- increased motivation and involvement of users,
- economies of specialization,
- permits exploitation of low-cost minicomputers,
- permits small increments to capacity,
- reduces communication costs,
- reduced interactive response time,
- increased reliability,
- increased predictability of costs.

3. Distributed systems

ADVANTAGES: economies of scale,
 increased efficiency,
 demand smoothing,
 incorporation /user - technical
 expertise/,
 integration of information proces-
 sing,
 integrati-~~n~~ of organizational ac-
 tivities,
 greater user control,
 reduced communication cost,
 reduced response time,
 reliability.

HAZARDS : creeping escalation,
 hidden costs,
 duplication and incompatibility,
 incompetent design,
 suboptimization.

Distributed systems are not simple extensions of monolithic systems.

There are computer systems /special purpose or general/ which are operating in a noncentralized manner. Packet-switching computer communication networks are examples of distributed systems.

In such systems two kinds of resources must be considered:

1. system resources /multiplexing is required/
2. user resources.

An example for 2. is distributed data-base sharing.

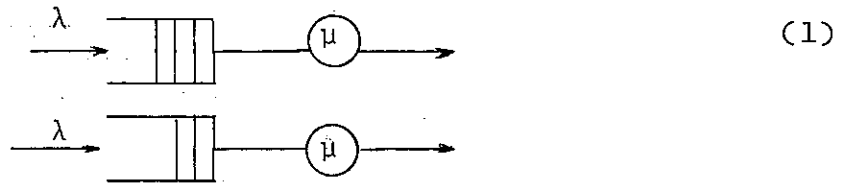
One of the problems when we design distributed computer systems is the scheduling of jobs among processors in the system.

The objective is to achieve system balance, with a resultant performance increase, by automatically shifting jobs from heavily loaded processors to lightly loaded processors in the system.

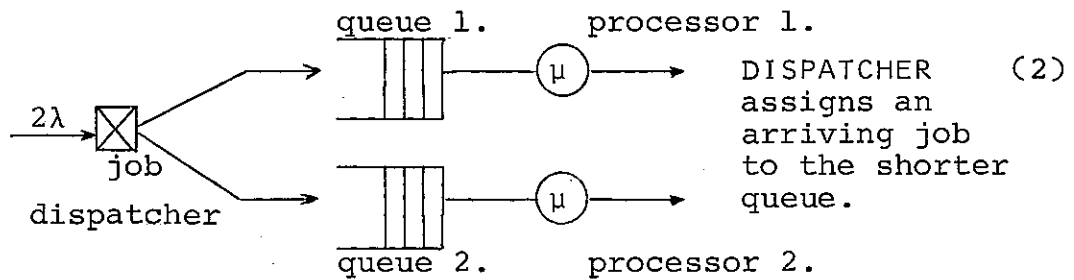
Load balancing can be done statistically, or dynamically as the load and the state of the system changes. Dynamic load balancing have taken two different approaches. One is a combinatorial optimization problem. Jobs are reassigned dynamically by monitoring the state of the system.

The other approach is to develop queuing models to analyze the performance of the systems incorporating simple job routing policies that automatically shift jobs.

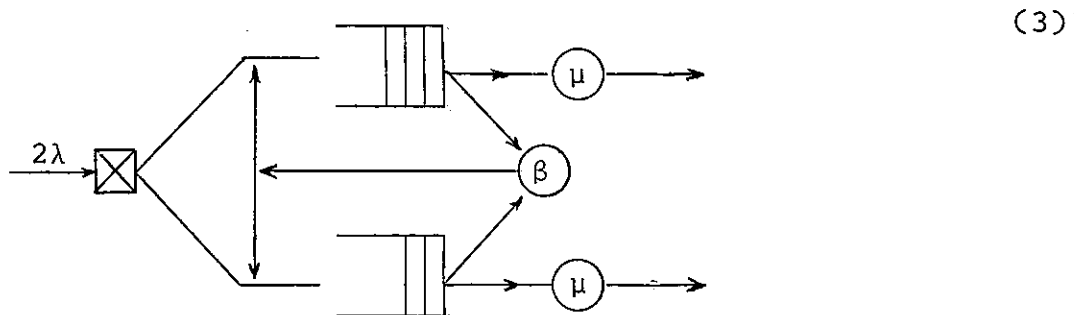
TWO PROCESSOR DISTRIBUTED COMPUTER SYSTEM
with load balancing



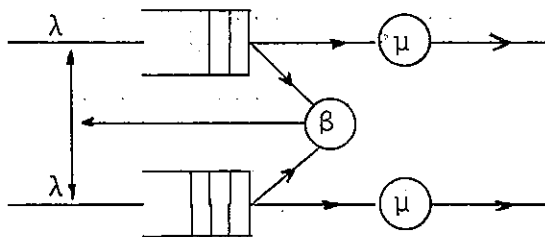
Join random queue without channel transfer.



Join shorter queue without channel transfer.

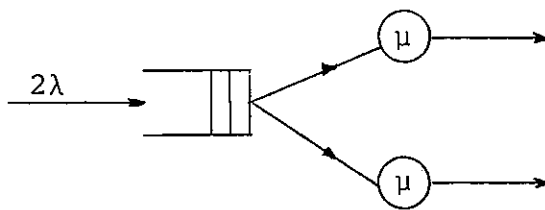


Join shorter queue with channel transfer.



(4)

Join random queue with channel transfer.



(5)

Instantaneous channel transfer.



(6)

Single fast processor system.

$$P_{ij}(t) = P\{X_1(t) = i, X_2(t) = j\}$$

$X_i(t)$ is the number of jobs in queue i . The KOLMOGOROV equations can be written in the ordinary way and the steady state equations can be used for the following characteristics. Utilization of the processors (U): $U = \lambda/\mu$,

Average queue length (L),

Mean job turn around time (T) which by Little law $\lambda T = L$ is in.

In case $\mu = 1$ the following values of T .

λ Model	0.1	0.2	0.5	0.9
1.	1.11	1.25	2.00	10.00
2.	1.02	1.06	1.42	5.47
3.	1.02	1.05	1.39	5.35
4.				
5.	1.01	1.04	1.33	5.26
6.	0.56	0.63	1.00	5.00

RESOURCE MANAGEMENT

1. In a single computer installation the resource management task is normally performed by the operating system and by the operator.

Relationship will exist between workload /input/ and computer performance, which depends on resource strategies. Resource management strategy allows intervention

on some parameters, by which one can tune the system.

w - computer workload
y - performance
a - parameters
 $g(\underline{a}, w)$ - cost function

$$y = \underline{P}(m, w)$$

$$g(\hat{a}, w) = \inf_{a \in M} g(a, w)$$

2. Resource management in computer network

$$y_i = P_i(a_i, w_i, u_i)$$

u_i - interaction variable
 /e.g. amount of ex-
 change workload at i /
 would reflect network con-
 figuration and routing
 algorithmus

$$g_i = g_i(a_i, w_i, u_i)$$

$$i = 1, 2, \dots, n$$

$$g(\hat{a}, \underline{w}, \underline{u}) = \inf_{a \in \tilde{M}} g(\underline{a}, \underline{w}, \underline{u})$$

There are two ways to overcome the conflict between global and local optimum control on resources:

- a/ Entrusting the global resource management task to just one decision making unit within the network /„centralized approach"/.
- b/ New decision-making unit /„coordinator"/ whose goal is hierarchically coordinate the local computer systems.

FILE ALLOCATION IN AN INFORMATION NETWORK

Transactions with the multiple-located file give rise

- a/ query traffic to a simple copy
- b/ update traffic to every copy

$$a_i = \begin{cases} 1 & \text{file is in } i\text{-th mode} \\ 0 & \text{otherwise} \end{cases} \quad [\text{CASEY'S MODEL}]$$

Costs: communication costs for query and update file storage costs.

d_{jk} - communication cost,

σ_k - storage cost,

I - the index set having a file copy

$$C_{\text{tot}} = \sum_{j=1}^n \left(\sum_{k \in I} r_j d_{jk} + q_j \min_{k \in I} d_{jk} \right) + \sum_{k \in I} \sigma_k$$

q_i - volume of query

r_i - volume of update

HIERARCHICAL ROUTING AND FLOW CONTROL POLICY (for packet switched networks)

Management of traffic flow in a computer network can be

divided into two main tasks:

- a/ message routing,
- b/ flow control.

a/ Message delay or cost is a commonly used parameter to design routing policy.

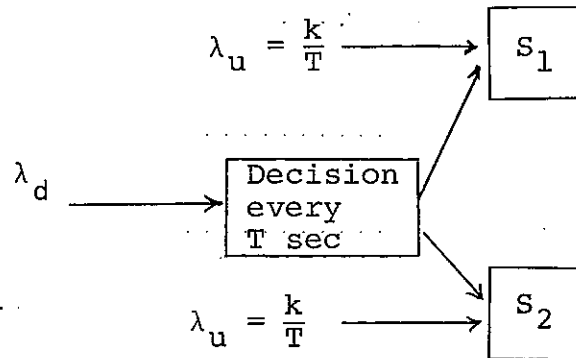
b/ Buffer size is the parameter for flow control algorithms.

It is desirable to study them jointly. Channel traffic intensity as a common parameter [it is closely related to delay and queue length] is used to jointly optimize the routing and flow control.

ROUTING UPDATES IN PACKET SWITCHED NETWORKS

Routing decisions in packet switched networks are often based upon congestion updates passed around the network. There is a natural conflict in the selection of the update period T :

- make T small to have decisions on recent information,
- make T large to minimize the overhead.



Poisson stream of data traffic with rate λ_d is assumed. Decisions are made every T seconds as to which server to route to, based on which server had the smallest queue at the beginning of the T -period. All packets are routed to the chosen server for the entire T -period.

The overhead /updating the queue information every T seconds/ shall be simply modeled as an additional Poisson stream with rate $\lambda_u = \frac{k}{T}$.

All service times are mutually independent exponential with $\frac{1}{\mu}$. It is assumed

$$\lambda_d + 2\lambda_u < 2\mu.$$

The λ_u -streams represent update packets to accomplish node-to-node transfer of congestion information accross the two links S_1 and S_2 .

Other representations of the input process and the update overhead can be visualized /updates arrive deterministically, preemptive priority, different service times/.

REFERENCES

- [1] J.C. EMERY: Managerial and economic issues in distributed computing. Information Processing, 77. 945-955.
- [2] G.Le LANN: Distributed systems -- toward a formal approach. Information Processing 77., 455-460.
- [3] M. MACKAWA: Interprocess communication in a highly diversified distributed systems. Information Processing 77., 149-154.
- [4] F.E. TAYLOR: The relative merits of distributed computing systems. International Computing Symposium, 1977., 357-365. N.M.P., 1977. Ed. E. Morlet, D.Ribbens.
- [5] G. BUCCI, S.GOLINELLI: A distributed strategy for resource allocation in information networks. Int. Computing Symposium, 1977., 345-356.
- [6] C. HEWITT, M.BAKER: Actors and continous functionals. Formal Description of Programming Concepts, 1978., 367-390. N.M.P.C. 1978., Ed. E.J. Nenhold.