РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ИНФОРМАЦИИ
(Вычислительные системы)
1984 год Выпуск 105

УДК 681.324

АЛГОРИТМ ПРЕДОТВРАЩЕНИЯ ДЕДЛОКОВ ПРИ ПЕРЕДАЧЕ СООБЩЕНИЙ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ И В СЕТЯХ

Э.А. Монахова

Одна из основных задач управления потоком сообщений [1,2] в вычислительных системах (ВС) и сетях - исключение возможности возникновения тупиковых состояний (дедлоков, блокировок пути). В ряде сетей и систем (ARPANET [3], GMDNET [4], SBNET [5] и др.) средства предотвращения дедлоков включены в программное обеспечение. При организации межмашинных взаимодействий в ВС с программируемой структурой [6] требуется эффективное решение проблемы обеспечения беступиковости передач сообщений (пакетов) между взаимодействующими элементарными машинами (ЭМ).

Тупиковые состояния при передаче пакетов (store-and-forward deadlock [2,3,5]) возникают вследствие переполнения буферных пулов ЭМ из-за ограниченности объемов буферных пулов и неупорядоченного их использования, когда поступающий в сеть поток превышает допустимый. В случае дедлока в системе образуется цикл ЭМ (в простейшем случае из двух соседних) с полностью заполненными буферными пулами, каждый из которых содержит пакеты, предназначенные для передачи следующей ЭМ цикла. В результате ни один из пакетов не может быть передан или принят.

Известные алгоритмы предотвращения дедлоков имеют ряд недостатков: 1) накладывают ограничения на возможные перемещения пакетов [7,8]; 2) задают сложное структурирование буферного пула ЭМ [4,7-8]; 3) предполагают увеличение размера пула в каждой ЭМ с ростом системы [4,7-8]; 4) требуют выделенной ЭМ с большим размером пула - "стока" пакетов с возможной потерей (уничтожением) их в системе [5].

Цель настоящей работы - предложить децентрализованный алгоритм предотвращения дедлоков, который свободен от перечисленных недостатков и может быть использован как в ВС с программируемой структурой, так и в вычислительных сетях.

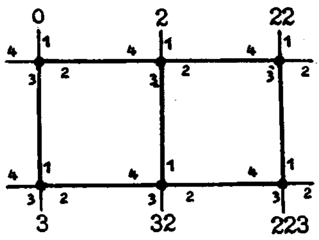
Специфика ВС с программируемой структурой - возможность одновременного решения различных задач на выделенных для них подмножествах ЭМ (быть может, и пересекающихся) - виртуальных подсистемах [6]. При этом весь буферный пул ЭМ делится между различными подсистемами на непересекающиеся буферные пулы подсистем. Подсистема Q_s (s - идентификатор подсистемы) состоит из требуемого задачей числа ЭМ, равного $|Q_s|$, соединенных линиями связи так, чтобы обеспечивались межмашинные взаимодействия, обусловленные алгоритмом решаемой задачи.

Алгоритм использует задание MD(z,m)-адресации (модифицированной D(z,m)-адресации) [9] ЭМ виртуальных подсистем ВС. Адрес $A_i = A_i^1 A_i^2 \ldots A_i^l$, $l \le m$, каждой ЭМ$_i$, $1 \le i \le |Q_s| - 1$, подсистемы Q_s представляет собой последовательность номеров выходных полюсов ЭМ подсистемы, лежащих на кратчайшем пути от корневой ЭМ$_0$ до ЭМ$_i$. В качестве корневой ЭМ$_0$ в подсистеме Q_s может быть выбрана любая ЭМ, адрес которой принимается равным нулю: $A_0 = 0$. Машины, адреса которых имеют одинаковую часть $A_i^1 A_i^2 \ldots A_i^{r-1}$ длины $r - 1$, $2 \le r \le m+1$, образуют связное подмножество ЭМ подсистемы, называемое адресной подсистемой $R_r(A_i^1 A_i^2 \ldots A_i^{r-1})$ яруса r.

Путевая функция $G_i(A_j)$ [9] определена в каждой ЭМ$_i$ подсистемы и ставит в однозначное соответствие каждому адресу A_j ЭМ подсистемы номер выходного полюса, принадлежащего кратчайшему пути по дереву путей от ЭМ$_i$ до адресной подсистемы $R_{r+1}(A_i^1 \ldots A_i^{r-1} A_j^r)$, к которой принадлежит ЭМ$_j$, где

$$r = \min \{k / A_i^k \ne A_j^k,\ 1 \le k \le m\}. \quad (1)$$

Путевая процедура (алгоритм маршрутизации), используя таблицы путевых функций, определяет трассы передачи сообщений между передающими и принимающими ЭМ подсистемы. По адресу A_j приемника, содержащемуся в сообщении, и адресу A_i ЭМ$_i$, исполняющей путевую процедуру, определяется номер выходного направления, по которому следует передать поступившее в ЭМ$_i$ сообщение. На рисунке изображено задание MD(z,m)-адресации для виртуальной подсистемы ВС, состоящей из 6-ти ЭМ, и путевой функции для ЭМ$_i$ с адресом $A_i = 2$.



$G_i(0) = 4$; $G_i(3) = G_i(32) = 3$;
$G_i(22) = G_i(223) = 2$; $G_i(2) = 0$

Заметим,что $MD(z,m)$-адресация является развитием иерархической адресации [10], предложенной для вычислительных сетей, и может быть реализована вместе с путевыми процедурами в вычислительных сетях.

Алгоритм предотвращения дедлоков основан на следующем задании системы приоритетов передаваемых пакетов.

Обозначим через P_i множество всех пакетов, находящихся в настоящий момент в буферном пуле ЭМ$_i$, $0 \le i \le |Q_s| - 1$. В это множество не входят пакеты, адресованные ЭМ$_i$, поскольку за конечное время они освобождают буфера и в дальнейших передачах не участвуют. Каждый пакет $p \in P_i$, в зависимости от адреса A_i ЭМ$_i$, исполняющей путевую процедуру, и адреса A_j своего приемника ЭМ$_j$($A_i \ne A_j$), принадлежит к одному из трех возможных классов $P_i = \{P_i^1, P_i^2, P_i^3\}$, т.е. в системе существуют пакеты классов I,2 и 3.

Наивысший приоритет имеют пакеты класса I. Условие принадлежности p к классу P_i^1:

$$r > l_i, \qquad (2)$$

где r - разряд рассогласования адресов A_i и A_j, определяемый по (I), l_i - расстояние ЭМ$_i$ до корневой ЭМ$_0$ подсистемы. Класс I составляют пакеты, находящиеся в адресной подсистеме своего приемника, следующий шаг которых, определенный путевой процедурой,направлен вниз от корневой ЭМ$_0$ по дереву путей, порождаемому заданной адресацией. Пакеты класса I могут занимать все буфера пула ЭМ. Если через δ_i обозначить число пустых буферов в пуле ЭМ$_i$, то $\delta_i > 0$ - условие принятия в пул пакета класса P_i^1.

Класс 2 составляют пакеты, следующий шаг которых по пути к приемнику направлен (или перенаправлен) вверх к корневой ЭМ$_0$ по дереву путей, порождаемому заданной адресацией.Условием принадлежности p к классу P_i^2 является равенство

$$G_i(A_j) = G_i(A_0). \qquad (3)$$

Условие принятия в пул.ЭМ$_i$ пакета класса P_i^2: $\delta_i > 1$.

Остальные пакеты, т.е. пакеты, для которых не выполнено ни одно из условий (2) и (3), относятся к классу 3 (P_i^3). Пакеты указанного класса принимаются в пул $ЭМ_i$ при $\delta_i > 1$ с соблюдением следующего требования: в случае занятия предпоследнего пустого буфера в пуле и при отсутствии в пуле пакетов других классов пакет класса P_i^3 должен быть перенаправлен в корневую ЭМ подсистемы и превращен тем самым в пакет класса P_i^2.

Основное требование алгоритма - пакеты классов 2 и 3 всегда оставляют последний пустой буфер пула для пакета класса I.

Для предотвращения ситуации дедлока в системе алгоритм использует процедуру перенаправления некоторых пакетов в корневую ЭМ подсистемы. Все пакеты, находящиеся в корневой ЭМ (в частности, перенаправленные пакеты, достигшие корневой ЭМ), и пакеты, идущие из корневой ЭМ к своим приемникам, принадлежат классу I наивысшего приоритета - таково свойство MD(z,m)-адресации. Таким образом, перенаправленные пакеты, достигшие корневой ЭМ, будут доставлены своим адресатам за конечное время, что следует из выполнения основного требования алгоритма и будет доказано ниже.

Заметим, что в каждой $ЭМ_i$ подсистемы есть только одно выходное направление, определяемое значением путевой функции $G_i(A_0)$, которое ведет к корневой $ЭМ_0$. По нему и будут перенаправляться в случае необходимости пакеты класса P_i^3. Из определения пакетных классов следует, что между любыми двумя соседними $ЭМ_i$ и $ЭМ_j$ подсистемы возможна передача пакетов классов P_i^1 и P_i^3 в $ЭМ_j$ и класса P_j^2 в $ЭМ_i$, если $l_i < l_j$, или передача пакетов класса P_i^3 в $ЭМ_j$ и класса P_j^3 в $ЭМ_i$, если $l_i = l_j$. Рассмотрим три различных метода перенаправления пакетов до корневой ЭМ.

I. Однократное перенаправление. При однократном перенаправлении пакет перенаправляется к корневой $ЭМ_0$ один раз в той ЭМ, в которой возникла необходимость перенаправления.

Пусть соседние $ЭМ_i$ и $ЭМ_j$ соединены линией связи, по которой возможна передача пакетов класса P_i^3 в $ЭМ_j$ и P_j^2 в $ЭМ_i$. Пусть пакет p класса P_i^3 передается в $ЭМ_j$, где также является пакетом класса P_j^3. В результате однократного перенаправления пакет p возвращается в $ЭМ_i$, и путевая процедура, определяющая номер выходного направления по адресу его приемника, снова направляет p в $ЭМ_j$. Если здесь опять создалась ситуация для его перенаправления, то он снова попадет в $ЭМ_i$ и т.д. Таким образом, при однократном перенаправлении создается возможность зацикливания пакета p до тех пор, пока в ЭМ$_i$ не появится ситуация для перенаправления p или, наоборот, в ЭМ$_j$ эта ситуация не исчезнет.

2. Многократное перенаправление. При таком перенаправлении первый раз перенаправленный пакет получает статус "перенаправленного" пакета и выводится из ЭМ$_i$ по выходному направлению с номером $G_i(A_0)$. В каждой транзитной ЭМ$_i$ путевая процедура, получив на вход q "перенаправленный" пакет, сравнивает номер входа с $G_i(A_j)$. Если $G_i(A_j) = q$, то пакет принадлежит к классу P_i^2 и номер его выходного направления равен $G_i(A_0)$. Если $G_i(A_j) \neq q$, то с p снимается статус "перенаправленного" пакета, определяется номер его выходного направления $G_i(A_j)$ и принадлежность к одному из классов P_i. При таком перенаправлении зацикливания пакета исключаются.

3. Полное перенаправление отличается от способа многократного перенаправления тем, что в каждой транзитной ЭМ$_i$ по пути в корневую ЭМ "перенаправленный" пакет независимо от номера входа q всегда принадлежит к классу P_i^2 и номер его выходного направления равен $G_i(A_0)$. При этом удается предотвратить ситуацию дедлока и зацикливание пакетов в системе ценой удлинения пути некоторых пакетов до приемника без введения "глобальных часов" в системе, как в [II], и без подсчета числа перенаправлений, как в [5].

Описанный ниже протокол реализации алгоритма предотвращения дедлоков i-й машинной подсистемы, $0 \le i \le |Q_m| - 1$, соответствует случаю использования однократных перенаправлений пакетов в системе.

Пусть p – пакет, стоящий на входе в ЭМ$_i$.

I. Если p адресован ЭМ$_i$, т.е. $A_i = A_j$, то ЭМ$_i$ принимает его в пул при $\delta_i > 0$.

2. Если p не адресован ЭМ$_i$, т.е. $A_i \neq A_j$, то, исполняя путевую процедуру над p, ЭМ$_i$ определяет его принадлежность к одному из трех классов P_i.

3. При $\delta_i > 1$ ЭМ$_i$ принимает в буферный пул пакет любого класса. Если после этого становится $\delta_i = 1$ и все пакеты в пуле принадлежат классу P_i^3, то один из них перенаправляется к корневой ЭМ (превращаясь в пакет класса P_i^2).

4. При $\delta_i = 1$ ЭМ$_i$. принимает в пул только пакет класса P_i^1.

Алгоритм определяет следующие ограничения на структуру заполнения буферного пула.

ЛЕММА. Каждая ЭМ виртуальной подсистемы в любой момент времени содержит в буферном пуле по крайней мере одну из следующих комбинаций: I) два пустых буфера; 2) пустой буфер и пакет класса I; 3) пустой буфер и пакет класса 2; 4) пакет класса I и пакет класса 2; 5) два пакета класса I.

ДОКАЗАТЕЛЬСТВО. По алгоритму (п.3) при числе пустых буферов в пуле $\delta > 1$ ЭМ принимает пакет любого класса, т.е. при заполнении пула с $\delta > 2$ никаких ограничений на класс пакетов не накладывается. Ясно, что в пуле будет по крайней мере два пустых буфера (случай I).

Пусть теперь $\delta = 2$ и ЭМ принимает в пул пакет р (δ становится равным единице).

Если р - пакет класса I, то имеет место случай 2.

Если р - пакет класса 2, то имеет место случай 3.

Если р - пакет класса 3, и все другие пакеты в буферном пуле также принадлежат классу 3, то после перенаправления пакета (п. 3 алгоритма) имеет место случай 3 ; если же среди других пакетов пула есть принадлежащие к классу I или классу 2, то имеет место случай 2 или случай 3 соответственно. Таким образом, при δ = I имеет место или случай 2, или случай 3 .

Далее, при δ = I ЭМ может принять в пул только пакет класса I (п.4 алгоритма), т.е. могут иметь место еще два случая: случай 4 и случай 5 . Лемма доказана.

ТЕОРЕМА. Если передача пакетов в вычислительной системе происходит по предложенному алгоритму, то в системе гарантировано отсутствие дедлоков.

ДОКАЗАТЕЛЬСТВО проведем методом от противного. Пусть в системе имеется дедлок. Значит [5] существует цикл длины $k \le |Q_s|$ из элементарных машин подсистемы $Q_s\{ЭМ_i^*\} = \{ЭМ_1^* < ЭМ_2^* < \ldots < ЭМ_k^* < ЭМ_1^*\}$, буферные пулы которых не содержат пустых буферов и пакетов, прибывших в свой приемник, и, кроме того, пакеты в каждой ЭМ цикла ожидают, когда пакеты следующей ЭМ цикла освободят для них буфера. Поскольку нет пустого буфера в пуле каждой $ЭМ^*_i$, то по лемме любая ЭМ цикла в каждый момент времени содержит по крайней мере один пакет класса I. Но по определению, следующий шаг пакета класса I всегда направлен вниз от корневой $ЭМ_0$ подсистемы, т.е. если $ЭМ^*_{i-1}$ предшествует $ЭМ^*_i$ в цикле, а $ЭМ^*_{i+1}$ следует за $ЭМ^*_i$ в цикле, то должно выполняться: $l_i < l_{i+1}$ для всех $i = 1,2,\dots,k-1$ и $l_i < l_{(i+1)\bmod k}$ для $i = k$, где l_i – расстояние $ЭМ^*_i$ до корневой $ЭМ_0$ подсистемы. Взяв транзитивное замыкание, получаем,что одновременно должно выполняться $l_1 < l_k$ и $l_k < l_1$. Полученное противоречие доказывает, что никакой пакет класса I не может находиться в состоянии дедлока.

Покажем теперь, что никакой пакет класса 2 не может находиться в состоянии дедлока. Поскольку последние незанятые буфера в пуле недоступны для пакетов класса 2, рассмотрим цикл $\{ЭМ^*_i\}$ машин подсистемы, каждая из которых содержит в пуле по одному пустому буферу. По лемме каждая ЭМ цикла содержит в пуле в любой момент времени или пакет класса I или пакет класса 2. Но если в буферных пулах ЭМ, составляющих цикл, есть хотя бы один пакет класса I, то он может занимать пустой буфер соседних ЭМ и дедлока в этом случае нет. Поэтому рассмотрим случай, когда в каждой ЭМ цикла нет пакетов класса I, а есть обязательно хотя бы один пакет класса 2. По определению,следующий шаг пакета класса 2 всегда направлен вверх к корневой $ЭМ_0$ подсистемы.Поэтому должно выполняться: $l_i > l_{i+1}$ для всех $i = 1,2,\dots,k-1$ и $l_i > l_{(i+1)\bmod k}$ для $i = k$. Аналогично первому случаю получим, что $l_1 > l_k$ и $l_k > l_1$ одновременно. Полученное противоречие доказывает, что никакой пакет класса 2 не может находиться в состоянии дедлока.

Покажем теперь, что пакеты класса 3 не могут находиться в состоянии дедлока. В этом случае рассмотрим цикл ЭМ с пулами, заполненными полностью пакетами класса 3 при одном пустом буфере.Но по п.3 алгоритма один пакет класса 3 в пуле каждой ЭМ цикла должет быть превращен в пакет класса 2, т.е. перенаправлен по выходному направлению, не совпадающему с направлением пакетов класса 3. Таким образом, рассматриваемый цикл просто не может существовать в системе, в которой передача пакетов происходит по предложенному алгоритму. Теорема доказана.

Предложенный в настоящей работе децентрализованный алгоритм гарантирует отсутствие дедлоков при передаче сообщений независимо в каждой подсистеме. Алгоритм не накладывает ограничений на возможные перемещения пакетов. Приоритетное обслуживание выделенных пакетных классов осуществляется посредством минимального структурирования буферного пула ЭМ. Минимально необходимый размер буферного пула в каждой ЭМ подсистемы равен двум буферам.

Литература

I. ЯКУБАЙТИС Э.А. Архитектура вычислительных сетей.-М.: Статистика, 1980. - 279 с.

2. ЛАЗАРЕВ В.Г., ЛАЗАРЕВ Ю.В. Динамическое управление потоками информации в сетях связи. -М.: Радио и связь, 1983. - 216 с.

3. KLEINROCK L. Principles and lessons in packet communications.- Proc.IEEE,1978,v.66,p.1320-1329.

4. GIESSLER A., HAENLE J., KOENIG A., PADE E. Free buffer allocation. - An investigation by simulation. - Comput.Networks, 1978,v.2, p.191-208.

5. GELERNTER D. A DAG-based algorithm for prevention of store-and-forward deadlock in packet networks.- IEEE Trans.Comput., 1981,v.C-30,N 10,p.709-715.

6. КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Вычислительные системы с программируемой структурой. - Электронное моделирование, 1979,№I, с. 42-52.

7. TOUEG S., ULLMAN J. Deadlock-free packet switching networks.-In: Proc.ACM Symp.Theory Comput.,1979,p.89.

8. MERLIN P., SCHWEITZER P. Deadlock avoidance in store-and-forward networks.- In: Store-and-forward deadlock.- IEEE Trans. Commun.,1980,v.COM-28,p.345-360.

9. КОРНЕЕВ В.В., МОНАХОВ О.Г. О децентрализованном распределении заданий в вычислительных системах с программируемой структурой. -Электронное моделирование, 1981, №6, с. 15-22.

10. ДЭВИС Д., БАРБЕР Д., ПРАЙС У., СОЛОМОНИДЕС С. Вычислительные сети и сетевые протоколы. -М.: Мир, 1982. - 563 с.

II. TOUEG S. Deadlock-and-livelock-free packet switching networks.- In: Proc.ACM Symp.Theory Comput.,1980,p.94-99.

Поступила в ред.-изд.отд.
16 апреля 1984 года