УДК 519.689.1

А. А. Переймибіда

ПРО ДЕЯКІ ПІДХОДИ ДО РОБОТИ З РОЗРІДЖЕНИМИ МАТРИЦЯМИ ІЗ ГРУПОВАНИМИ НЕНУЛЬОВИМИ ЕЛЕМЕНТАМИ

1. Постановка задачі. При програмній реалізації методів розв'язування задач математичної фізики виникає потреба працювати з матрицями великих розмірностей. Програмування під DOS накладає низку обмежень на використання пам'яті.

Наступні підходи можна використовувати для розв'язання перелічених проблем: проблема одного сегмента може бути розв'язана через динамічний розподіл пам'яті; використання Extended або Expanded пам'яті дасть змогу використовувати всю доступну оперативну пам'ять; тимчасове збереження інформації на диск з наступним підвантаженням необхідної інформації, дозволить використовувати пам'ять у межах доступного дискового простору та оперативної пам'яті (далі: *доступної пам'яті*). Засобами ООП можна приховати основний недолік таких підходів: різні реалізації для підтримки різних типів пам'яті.

Сучасні операційні системи (наприклад, Windows 95) підтримують реалізацію віртуальної пам'яті [1]. Часто у задачах математичної фізики виникають розріджені матриці. Виникає питання: чи завжди доцільно у таких випадках використовувати стандартну реалізацію віртуальної пам'яті?

2. Робота з віртуальною пам'яттю. Якщо ми хочемо замовити великі об'єми пам'яті, а використовуємо лише деякі елементи – слід використати віртуальну пам'ять. При виділенні (Commit) деякого об'єму пам'яті, ОС розподіляє для процесу адресний простір потрібного розміру. Але об'єм у пам'яті чи на диску не займається. Далі, у межах виділеної, можна резервувати (Reserve) пам'ять, з якою вже можна працювати. Після цього розмір доступної пам'яті у системі зменшується на відповідну величину. Мінімальний розмір резервованої пам'яті – 1 сторінка ОС (у Windows 95 – 4k).

3. Приклад реалізації. У системах програмування (розглядається реалізація у Delphi), при описі великих масивів у віртуальній пам'яті резервується необхідний об'єм. Тобто на диску, у

4. Приклад використання для діагональних матриць. Розглянуто випадок, що виник при розв'язуванні задач математичної фізики. Побудова програмного комплексу для розв'язування телеграфного рівняння (наприклад для випадку розглянутого у [2]) зумовила необхідність використовувати великі масиви. Реалізація під DOS використовувала перераховані у пункті 1 методи обходу обмежень системи при використанні пам'яті.

Аналіз структури отримуваних матриць показав їх спеціальну циклічну стрічкову структуру (рис. 1). Кількість стрічок, що будуть заповнені, не піддається попередньому визначенню.

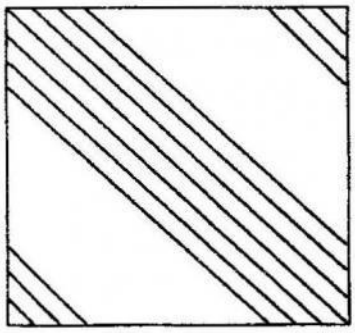

Рис. 1. Стрічковість отримуваних матриць

Ефективним використання описаного вище класу *VirtualArray* буде лише тоді, коли розмірність матриці перевищить 512x512. Для нашої задачі було реалізовано нащадок цього класу: *VirtualBandArray:* замовляється масив *VirtualArray* розміром $Nx(N+2(N-1))$, а доступ здійснюється за зміщеною системою координат $\begin{cases} i' = i \\ j' = j - i + N \end{cases}$. Оскільки в *VirtualArray* резервуються не нульові елементи, то в об'ємі необхідної пам'яті ми втратимо лише на системні потреби. Недоліком є лише те, що розходується адресний простір об'ємом $(Nx(N+2(N-1))x\,8)$. Виходячи з величини адресного простору (2-4Гб), можна вважати, що даний недолік не є суттєвим.

Таким чином, пряме використання стандартних можливостей не завжди ефективне. Ми описали спеціальні прийоми для більш ефективної роботи з розрідженими матрицями у Windows-системах.

1. Калверт Ч. Delphi 2. Энциклопедия пользователя. К.: НИПФ "ДиаСофт Лтд.", 1996. - 736. 2. Хапко Р. С., Переймибіда А. А. Метод потенціалів розв'язування початково-крайових задач для телеграфного рівняння на площині. //Теорет. електротехніка. 1996. №53. С. 73-82.

файлі для підвантаження (swap file), займається необхідний об'єм пам'яті. При описі *a:array [1..5000,1..5000] of double*, необхідно мати для запуску програми 200 000 000 байт місця для змінної *a*.

Реалізовано клас VirtualArray, який дає змогу працювати з великими двовимірними масивами, але орієнтований на збереження розріджених матриць. Лише при збереженні ненульових елементів у системі виділяється необхідний об'єм пам'яті. Таким чином, з урахуванням службових затрат, для збереження нульової матриці, згаданої вище, необхідно 48 842 байт.

У табл. 1 наведено результати порівняння класичного та пропонованого підходу. Тестування проводились на комп'ютері з процесором Pentium 200 MMX (ОП: 32Мб) для масивів розмірністю 5000x5000. Нулі у тестах слідували один за другим по стрічках матриці.

Таблиця 1. Час доступу до матриць залежно від заповненості

Заповне-ність	Читання (хв.)	Перший запис (хв.)*	Наступні записи (хв.)	Економія (наближ., Мб)
0%	0:27	0:28	0:28	200
40%	0:49	0:38	1:16	120
70%	1:00	0:50	2:02	60
Класичний	0:39	0:42	3:08	-

Результати свідчать про ефективність реалізованого класу за рахунок економії об'єму використаної пам'яті. Запис/читання у розріджену матрицю відбувається тим швидше, чим більш розріджена матриця.

Зауважено, що даний підхід є не завжди ефективний. Результати, наведені у табл. 1, отримані для матриць із неперервним розміщенням ненульових елементів (стрічка за стрічкою). Якщо ж ці елементи не груповані (розміщені випадково), то, за рахунок того, що мінімальний об'єм пам'яті для виділення становить одну сторінку (4k), ми можемо і не отримати виграшу. Ефективне використання класу VirtualArray буде лише тоді, коли відстані між групами ненульових елементів є більші за 1 сторінку.

---

* Перший запис у масив, виділений у віртуальній пам'яті, відбувається значно швидше за рахунок того, що не потрібно узгоджувати дані, які записуються, з даними збереженими у файлі підвантаження.