DOI 10.37882/2223–2966.2020.11.33

ФОРМАЛЬНЫЙ ПОДХОД К ОПТИМИЗАЦИИ РАЗМЕЩЕНИЯ ДИНАМИЧЕСКИХ МАССИВОВ ДАННЫХ МЕТОДОМ ПРЕДВАРИТЕЛЬНОГО РЕЗЕРВИРОВАНИЯ БЛОКОВ СТАТИЧЕСКОЙ ПАМЯТИ ЭВМ

A FORMAL APPROACH TO OPTIMIZING THE PLACEMENT OF DYNAMIC DATA ARRAYS BY PRE-RESERVING BLOCKS OF STATIC COMPUTER MEMORY

M. Tomaev
A. Gamidi

Summary. The problem of optimizing the placement of user data in multi-level computer memory by preemptive caching is Formalized. Two equivalent approaches to the formulation are given, and the relationship between them is demonstrated. The research is a theoretical basis for creating technologies and tools for creating optimal software products.

Keywords: optimization, program, model, search, quality, caching, static, dynamic.

Томаев Мурат Хасанбекович
Доцент, Северо-Кавказский государственный горно-металлургический институт (государственный технологический университет), Владикавказ
Гамиди Артур Олегович
Аспирант, Северо-Кавказский государственный горно-металлургический институт (государственный технологический университет), Владикавказ
a.gamidi@yandex.ru

Аннотация. Формализуется задача оптимизации размещения пользовательских данных в многоуровневой памяти ЭВМ методом упреждающего кэширования. Приведены два эквивалентных подхода к формулировке, демонстрируется взаимосвязь между ними. Исследование является теоретической основой для создания технологии и средств создания оптимальных программных продуктов.

Ключевые слова: оптимизация, программа, модель, поиск, качество, кэширование, статическая, динамическая.

Введение

Проблема эффективного размещения данных, используемых программой, в многоуровневой оперативной памяти ЭВМ (статическая память, стек, динамическая память) особенно актуальна для высоконагруженных вычислительных систем. С развитием языков программирования, включением в их состав библиотек, автоматизирующих наиболее распространенные операции по размещению и обработке данных, сформировалась тенденция размещения большей части пользовательских данных в динамической памяти ЭВМ. Однако, при выборе стратегии размещения данных в динамически выделяемых блоках ОП имеет необходимо учитывать возможные проблемы с производительностью. При высокой загрузке ОП возрастает сложность поиска эффективных вариантов дефрагментации, что приводит к снижению скорости обработки запросов на выделение новых блоков менеджером динамической памяти и, соответственно, становится причиной падения производительности операционной системы и прикладных программных пакетов. Таким образом одним из способов повышения производительности сложных программных комплексов является поиск эффективных стратегий перемещения данных из динамической «кучи» (heap) в сегменты фиксированного размера: сегмент глобальных данных либо стек.

В следующих листингах 1 и 2 (используется язык C++) приводится пример упреждающего кэширования, позволяющего избежать обращения к менеджеру динамической памяти до тех пор, пока размер массива не достигнет 1000 элементов.

Листинг 1. Исходная версия.

```
double *x = new [UserSize]; // выделение блока
….
delete [] x; //освобождение
```

Листинг 2. Оптимизированная версия, использующая упреждающее кэширование.

```
double *x;
static double xCache[1000];
x = (UserSize > 1000? new [UserSize]: xCache);
…
If (x!= xCache) delete [] x; // освободение
```

В работе предлагается формализация метода упреждающего резервирования блоков динамической памяти. Демонстрируется оптимизационный характер задачи. Цель исследования — подготовка формальной базы для создания алгоритмов и инструментальных средств, автоматизирующих процессы поиска эффективных стратегий модификации исходного кода, позволяющих улучшить производительность проектируемых программных систем.

Формулировка оптимизационной задачи

Обозначения.

M — число динамических массивов данных;
V — верхняя граница доступного объема ОП;
W_i — средний размер i-го динамического блока данных;
U_i — средний размер вспомогательного статического массива используемого для упреждающего кэширования i-го динамического массива;
N_i — среднее количество операций выделения i-го динамического блока данных.

Математическая модель

В данной главе формулируется непрерывная оптимизационная задача (1), минимизирующая суммарное время выделения блоков динамической памяти, среднестатистический размер которых известен (W_i). Демонстрируется переход к эквивалентной задаче (5), максимизирующей суммарное время размещения данных в статическом сегменте данных.

Сформулируем оптимизационную задачу, снижающую суммарное время выделения динамических блоков, используемых для размещения массивов данных.

$$\begin{cases} F = \sum_{i=1}^{m} N_i \frac{W_i - U_i}{s} \to min; \\ \sum_{i=1}^{m} z_i W_i \le V; \\ \forall_i : 0 \le U_i \le W_i; \\ z_i = 0,1; \end{cases} \quad (1)$$

Преобразуем выражение в целевой функции так, чтобы левая сумма содержала только постоянные коэффициенты, а правая включала неизвестную z_i:

$$F = \sum_{i=1}^{m} N_i \frac{W_i}{s} - \sum_{i=1}^{m} N_i \frac{z_i U_i}{s} \to min; \quad (2)$$

Уберем левую сумму, т.к. она представляет собой постоянную, не влияющую на оптимум:

$$F = -\sum_{i=1}^{m} N_i \frac{z_i U_i}{s} \to min; \quad (3)$$

Изменим знак перед суммой, и, соответственно, тип целевой функции с min на max:

$$F = \sum_{i=1}^{m} N_i \frac{z_i U_i}{s} \to max; \quad (4)$$

Упростим выражение, убрав постоянный коэффициент $1/s$ и окончательно сформулируем оптимизационную модель в виде:

$$\begin{cases} F = \sum_{i=1}^{m} N_i z_i U_i \to max; \\ \sum_{i=1}^{m} z_i W_i \le V; \\ \forall_i : 0 \le U_i \le W_i; \\ z_i = 0,1; \end{cases} \quad (5)$$

Смысл целевой функции в модели (5) заключается в максимизации суммарного объема данных, размещенного в статических массивах упреждающего кэширования вместо динамической (более медленной) памяти операционной системы.

Заключение

Предложенный в работе подход к размещению данных позволит снизить частоту обращений и, соответственно, снизить вероятность неконтролируемого падения производительности сложных высоконагруженных систем. Предложенный формализм будет востребован при создании программных средств разработки оптимальных программных продуктов.

ЛИТЕРАТУРЫ

1. Побегайло А. П. Системное программирование в Windows. БХВ-Петербург, Спб.:2017
2. Уильямс Энтони. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ. ДМК-Пресс, М:2012.
3. Томаев М.Х., Панарин В. Е. Выбор оптимального класса памяти для данных, используемых в программах на языке «C++». Материалы XIV международной научно-технической конференции «ИТ-технологии: развитие и приложения. г. Владикавказ, 2013», с. 215.

4. Томаев М. Х. Использование оптимизационных моделей «экстремального программирования» в проектировании ПО. Выбор оптимальной стратегии макрозамен.IT-технологии: теория и практика. Материалы семинара. Владикавказ, 2017, стр. 39–55.
5. Томаев М.Х., Гамиди А. О. Формализация метода кэширования функций произвольного числа переменных. Наука и бизнес: пути развития. Научно-практический журнал. № 11(101), 2019. Стр. 75–79.

Северо-Кавказский государственный горно-металлургический институт