САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Архитектура ЭВМ

Отчет

по домашней работе №3

«Кэш-память»

Выполнил(а): Алексеев Иван Алексеевич

студ. гр. М3139

Санкт-Петербург

2020

Цель работы: закрепление материала по теме «кэш-память» путем решения задач по данной теме.

Условие задачи

Вариант 2.

Имеется следующее определение глобальных переменных и функций

Вариант	Глобальные переменные	Функции

2	`unsigned int size = 1024 * 1024;`
	`double x[size];`
	`double y[size];`
	`double z[size];`
	`double xx[size];`
	`double yy[size];`
	`double zz[size];`
		`void f(double w)`
		`{`
		`  for (unsinged int i=0; i<size; ++i)`
		`  {`
		`  x[i] = xx[i] * w + x[i];`
		`  y[i] = yy[i] * w + y[i];`
		`  z[i] = zz[i] * w + z[i];`
		`  }`
		`}`

Рисунок 1 – Условие задачи

Рассмотрим систему с L1 кэшем данных с ассоциативностью 4-way размером 32 КБ и размером строки 64 байта. Кэш L2 представляет собой 8-way ассоциативный кэш размером 1 МБ и размером строки 64 байта. Алгоритм вытеснения: LRU. Массивы последовательно хранятся в памяти, и первый из них начинается с адреса, кратного 1024.

Определите процент попаданий (число попаданий к общему числу обращений) для кэшей L1 и L2 для выполнения предложенной функции.

В ответе нужно представить два числа, равных % попаданий для L1 и L2 кэшей.

Решение задачи

Тип double занимает 8 байт. Так как у кэшей L1 и L2 размер строки одинаковый, то в обоих в одной строке кэша поместится ровно $\frac{64}{8} = 8$ значений типа double.

В кэше L1 поместится $\frac{32*2^{10}}{8*8} = 2^7 = 512$ строк кэша. Так как кэш L1 имеет $\frac{32*2^{10}}{64} = 512$ ячеек, и каждая группа состоит из 4 ячеек, всего мы имеем $\frac{512}{4} = 128$ групп.

Аналогично в кэше L2 поместится $\frac{1*2^{20}}{8*8} = 2^{14} = 16\,384$ строк кэша. Так как кэш L2 имеет $\frac{1*2^{20}}{64} = 16\,384$ ячеек, и каждая группа состоит из 8 ячеек, всего мы имеем $\frac{16\,384}{8} = 2\,048$ групп.

Так как массивы x, xx, y, yy, z, zz хранятся в памяти последовательно, и первый из них начинается с адреса, кратного 1024, то строки кэша x[i], xx[i], y[i], yy[i], z[i], zz[i], где i – это индекс от массива от 0 до size, будут пытаться влезть в одну и ту же группу кэша L1. (Так как в L1 всего 128 групп, то ячейки будут записываться в группы по модулю $128 * 64 = 8192$, но размер массива size кратен этому модулю). Аналогично будет происходить с кэшем L2, так как размер модуля его групп ($2048 * 64 = 131\,072$) является делителем размера массивов, то строки кэша x[i], xx[i], y[i], yy[i], z[i], zz[i] будут пытаться влезть в одну и ту же группу.

Давайте рассмотрим запросы, которые получит кэш за первые 8 итераций нашего цикла (Рис. 2). Минус – элемента в кэше нет, плюс – есть. За одну итерацию цикла произойдет 9 запросов в кэш (сначала мы вызываем xx[i], умножаем xx[i] на w, получившееся выражение запоминаем в регистре, запрашиваем x[i], складываем, снова запоминаем выражение регистром, запрашиваем x[i] и присваиваем ему выражение из регистра. На всё – три запроса. Аналогично для y[i] и z[i]). Заметим, что на первой итерации мы в кэше L2 запоминаем x[i], y[i], z[i], xx[i], yy[i], zz[i], а значит и любое другое выражение x[i + k], y[i + k], z[i + k], xx[i + k], yy[i + k], zz[i + k] при целом k от 0 до 7 содержится в L2, так как в строке кэша содержится восемь

значений типа double. (В кэше мы запоминаем не значение, а строку кэша). Однако для L1 это неверно. Как мы уже говорили выше, L1 – это кэш с ассоциативностью 4-way, и при запросе xx[i], x[i], yy[i], y[i], zz[i], z[i] он будет стараться закинуть их в одну группу. Однако уже после 6 запроса (как видно из Рис. 2) все четыре ячейки одной группы будут заполнены и уже на 7 запросе мы выкинем xx[i] (так как у нас алгоритм вытеснения LRU). На 8 шаге кэш выкинет x[i] и присвоит в пустое место z[i]. И так далее.

№	Запрос	L1	L2
1	xx[i]	-	-
2	x[i]	-	-
3	x[i]	+	+
4	yy[i]	-	-
5	y[i]	-	-
6	y[i]	+	+
7	zz[i]	-	-
8	z[i]	-	-
9	z[i]	+	+

***

№	Запрос	L1	L2
1+9k	xx[i+k]	-	+
2+9k	x[i+k]	-	+
3+9k	x[i+k]	+	+
4+9k	yy[i+k]	-	+
5+9k	y[i+k]	-	+
6+9k	y[i+k]	+	+
7+9k	zz[i+k]	-	+
8+9k	z[i+k]	-	+
9+9k	z[i+k]	+	+

***

№	Запрос	L1	L2
64	xx[i+7]	-	+
65	x[i+7]	-	+
66	x[i+7]	+	+
67	yy[i+7]	-	+
68	y[i+7]	-	+
69	y[i+7]	+	+
70	zz[i+7]	-	+
71	z[i+7]	-	+
72	z[i+7]	+	+

Рисунок 2 – Первые 8 итераций

Однако на 9 итерации цикла мы запросим xx[i+8], но этого элемента нет ни в L1, ни в L2, так как это уже часть другой строки кэша. Значит фактически у нас $\frac{1024*1024}{8} = 131\,072$ раз будет повторяться одна и та же ситуация, описанная выше.

Посчитаем процент попаданий для кэшей L1 и L2.

Для L1: $\frac{131\,072*8*3}{131\,072*8*9} * 100\% = 33{,}33\%$ (131 072 раза повторяется ситуация, где из $8 * 9 = 72$ запросов попадаем только $8 * 3 = 24$ раза.)

Для L2: $\frac{131\,072*7*6}{131\,072*8*6} * 100\% = 87{,}5\%$ (131 072 раза повторяется ситуация, где из $8 * 6 = 48$ запросов попадаем только $7 * 6 = 42$ раза. Заметим, что мы считаем запрос в L2, только когда строки кэша нет в L1)