

**Исследование и  
оптимизация кэша данных  
по количеству строк и  
степени ассоциативности**

Скрипаль Б.

Алексюк А.

# Задание

- Исследование и оптимизация кэша данных в зависимости от:
  - Размера строки
  - Степени ассоциативности
- Рабочая нагрузка – библиотека VLAS (библиотека для работы с матрицами)

# Постановка эксперимента

## Задача:

- Нахождение оптимальных параметров кэша, при котором количество попаданий будет максимальным.

## Параметры эксперимента:

- Тип кэша – кэш данных первого уровня;
- Размер кэша – 64КБ;
- Размер строки варьируется в диапазоне от 8 до 32 байт;
- Степень ассоциативности варьируется от 1 до 4;
- Рабочая нагрузка – типовые операции с матрицами, реализуемые при помощи библиотеки BLAS (умножение двух матриц);
- Размер матриц – 1000 на 1000.

# Используемые средства

- Утилита pin 3.0;
- Библиотека pinatrace.so;
- Библиотека BLAS
- Ubuntu 16.04 x86\_64
- Intel Core i7-4690

# Описание библиотеки pinatrace.so

- Данная библиотека предназначена для отслеживания адресов данных, с которыми происходит работа в процессе исполнения программы.
- Результат работы библиотеки – список команд, обращающиеся к памяти, тип операции (чтение или запись) и адрес ячейки памяти, к которой происходило обращение.

# Пример исходных данных

- 0x40001ee0: R 0xbfffe798
- 0x40001efd: W 0xbfffe7d4
- 0x40001f09: W 0xbfffe7d8
- 0x40001f20: W 0xbfffe864
- 0x40001f20: W 0xbfffe868
- 0x40001f20: W 0xbfffe86c
- 0x40001f20: W 0xbfffe870
- 0x40001f20: W 0xbfffe874
- 0x40001f20: W 0xbfffe878
- 0x40001f20: W 0xbfffe87c

# Организация эксперимента

1. Модификация библиотеки `pinatrace.so`;
2. Формирование нагрузки;
3. Запуск утилиты `pin` для моделирования работы кэша;
4. Анализ результатов.

# Формирование нагрузки

- Операции над матрицами размером 1000x1000 элементов с типом double.
- Матрицы организованы по столбцам.
- Библиотека BLAS, функция DGEMM (англ. General Matrix Multiply).
- Выполняемая операция:
  - $C := \alpha * A * B + \beta * C$
  - В нашем случае  $\alpha = 1$ ,  $\beta = 0$



# Модификация библиотеки pinatrace.so

- Помещение внутрь исходного кода библиотеки модели кэша данных;
- Вместо вывода адреса команды в файл – анализ того, попали ли мы в кэш или нет и обновление модели кэша и статистики;
- Вывод накопленной статистики в файл.

# Модификация библиотеки pinatrace.so

```
unsigned long num = *((unsigned long*)&addr) / line_size;  
unsigned int current_line = num % bank_line_num;
```

```
int hit_pos = -1;
```

```
for (unsigned int bank = 0; bank < bank_num; bank++) {  
    if (custom_cache[bank][current_line].addr == num) {  
        hit_pos = (int) bank;  
        break;  
    }  
}
```

```
if (hit_pos > -1) {
    hit++;
    custom_cache[hit_pos][current_line].usage_counter = counter;
} else {
    miss++;
    unsigned long min = ULONG_MAX;
    unsigned long usage;
    for (unsigned int bank = 0; bank < bank_num; bank++) {
        usage = custom_cache[bank][current_line].usage_counter;
        if (usage < min) {
            min = usage;
        }
    }
}
```

```
for (unsigned int bank = 0; bank < bank_num; bank++) {
    usage = custom_cache[bank][current_line].usage_counter;
    if (usage == min) {
        custom_cache[bank][current_line].addr = num;
        custom_cache[bank][current_line].usage_counter =
counter;
        break;
    }
}
}
}
counter++;
```

# Результаты

Ассциативност

ь	Длина стр.	Кол-во строк	Попадания	Промахи	Всего	% попаданий
1	8	8192	1934671165	1129079658	3063750823	63.147145
1	16	4096	2498154533	565596290	3063750823	81.539090
1	32	2048	2779769052	283981771	3063750823	90.730912
2	8	4096	2056689405	1007061418	3063750823	67.129787
2	16	2048	2560203063	503547760	3063750823	83.564337
2	32	1024	2811955363	251795460	3063750823	91.781464
4	8	2048	2056689763	1007061060	3063750823	67.129799
4	16	1024	2560195235	503555588	3063750823	83.564082
4	32	512	2811940870	251809953	3063750823	91.780991

# Результаты

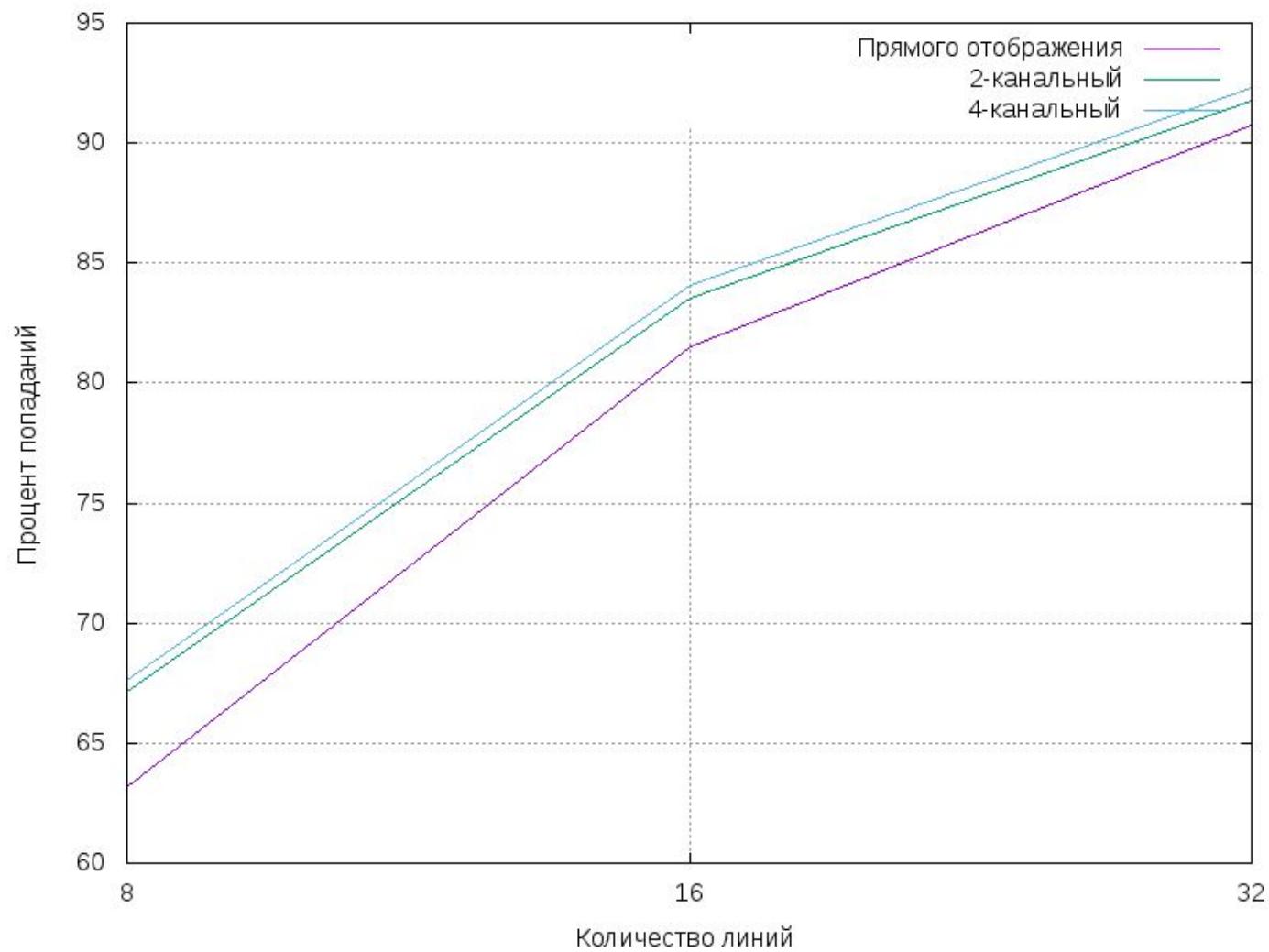
```
$ time ../../../../pin -t obj-intel64/pinatrace.so — ~/usedgemm  
time for C(1000,1000) = A(1000,1000) B(1000,1000) is 608.118 s  
Cache banks: 1, line_size: 8, lines: 8192, hit: 1934671165, miss:  
1129079658, all: 3063750823, hit percent: 63.147145
```

...

```
real 10m17.223s  
user 10m17.084s  
sys 0m0.080s
```

```
$ ./usedgemm  
time for C(1000,1000) = A(1000,1000) B(1000,1000) is 0.505199 s
```

# Обработка результатов



# Выводы

- С увеличением длины линии, количество попаданий возрастает
- Процент попадания при степени ассоциативности равной 2 и 4 практически совпадают
- Лучший результат: степень ассоциативности – 4 и длина линии 32 байта
- Оптимальный результат: степень ассоциативности – 2 и длина линии 32 байта