

Подсчет кэш промахов

Существующие инструменты

Valgrind – фреймворк для создания аналитических инструментов

- Содержит инструменты, которые позволяют отлаживать и профилировать программы
- Активно используется в большом количестве проектов и компаний
- **ПРЕДУПРЕЖДЕНИЕ: очень медленный**



Valgrind

- Определение ошибок работы с памятью (memcheck (default))
- Определение ошибок при работе с несколькими потоками (hellgrind)
- *Профилирование работы кэша и предсказания переходов (cachegrind)*
- Генерация графа вызовов (callgrind)
- Профилирование кучи (massif)

Цель

- Разобраться с проблемой $n=1024$ не «в уме», а «на мониторе»
 - Написать свой анализатор промахов
 - Сравнить результат с **cachegrind**

valgrind

- `sudo apt-get install valgrind`
- `g++ -g --std=c++11 main.cpp -o cache`
- `valgrind ./cache 1024`
- `valgrind --tool=cachegrind ./cache 1024`

Простой запуск

`$ valgrind ./cache 1023`

```
atolstikov@atolstikov-VirtualBox:/media/sf_2015_Algo/03_valgrind$ valgrind ./cache 1023
==7334== Memcheck, a memory error detector
==7334== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==7334== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info
==7334== Command: ./cache 1023
==7334==
n = 1023
timeSimple: 128.756
==7334==
==7334== HEAP SUMMARY:
==7334==    in use at exit: 0 bytes in 0 blocks
==7334==   total heap usage: 3 allocs, 3 frees, 12,558,348 bytes allocated
==7334==
==7334== All heap blocks were freed -- no leaks are possible
==7334==
==7334== For counts of detected and suppressed errors, rerun with: -v
==7334== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
atolstikov@atolstikov-VirtualBox:/media/sf_2015_Algo/03_valgrind$ ls
```

valgrind --tool=cachegrind ./cache 1023

```
atolstikov@atolstikov-VirtualBox:/media/sf_2015_Algo/03_valgrind$ valgrind --tool=cachegrind ./cache 1023
==7376== Cachegrind, a cache and branch-prediction profiler
==7376== Copyright (C) 2002-2013, and GNU GPL'd, by Nicholas Nethercote et al.
==7376== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info
==7376== Command: ./cache 1023
==7376==
--7376-- Warning: Cannot auto-detect cache config, using defaults.
--7376--      Run with -v to see.
n = 1023
timeSimple: 244.781
==7376==
==7376== I   refs:      50,117,664,317
==7376== I1  misses:      1,677
==7376== L1i misses:      1,659
==7376== I1  miss rate:      0.00%
==7376== L1i miss rate:      0.00%
==7376==
==7376== D   refs:      25,151,735,782 (23,854,955,462 rd + 1,296,780,320 wr)
==7376== D1  misses:      261,678,509 ( 261,472,481 rd +      206,028 wr)
==7376== L1d misses:      67,184,627 ( 66,987,079 rd +      197,548 wr)
==7376== D1  miss rate:      1.0% (      1.0% +      0.0% )
==7376== L1d miss rate:      0.2% (      0.2% +      0.0% )
==7376==
==7376== LL refs:      261,680,186 ( 261,474,158 rd +      206,028 wr)
==7376== LL misses:      67,186,286 ( 66,988,738 rd +      197,548 wr)
==7376== LL miss rate:      0.0% (      0.0% +      0.0% )
```

valgrind --tool=cachegrind ./cache

1024

```
atolstikov@atolstikov-VirtualBox:/media/sf_2015_Algo/03_valgrind$ valgrind --tool=cachegrind ./cache 1024
==7378== Cachegrind, a cache and branch-prediction profiler
==7378== Copyright (C) 2002-2013, and GNU GPL'd, by Nicholas Nethercote et al.
==7378== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info
==7378== Command: ./cache 1024
==7378==
--7378-- Warning: Cannot auto-detect cache config, using defaults.
--7378--      Run with -v to see.
n = 1024
timeSimple: 300.089
==7378==
==7378== I   refs:      50,263,925,560
==7378== I1 misses:      1,676
==7378== LLi misses:      1,658
==7378== I1 miss rate:      0.00%
==7378== LLi miss rate:      0.00%
==7378==
==7378== D   refs:      25,225,048,594 (23,924,683,465 rd + 1,300,365,129 wr)
==7378== D1 misses:      1,076,214,017 ( 1,075,491,517 rd +      722,500 wr)
==7378== LLd misses:      1,075,149,885 ( 1,074,951,953 rd +      197,932 wr)
==7378== D1 miss rate:      4.2% (      4.4% +      0.0% )
==7378== LLd miss rate:      4.2% (      4.4% +      0.0% )
==7378==
==7378== LL refs:      1,076,215,693 ( 1,075,493,193 rd +      722,500 wr)
==7378== LL misses:      1,075,151,543 ( 1,074,953,611 rd +      197,932 wr)
==7378== LL miss rate:      1.4% (      1.4% +      0.0% )
```


valgrind --tool=cachegrind ./cache

1025

```
atolstikov@atolstikov-VirtualBox:/media/sf_2015_Algo/03_valgrind$ valgrind --tool=cachegrind ./cache 1025
==7387== Cachegrind, a cache and branch-prediction profiler
==7387== Copyright (C) 2002-2013, and GNU GPL'd, by Nicholas Nethercote et al.
==7387== Using Valgrind-3.10.0.SVN and LibVEX; rerun with -h for copyright info
==7387== Command: ./cache 1025
==7387==
--7387-- Warning: Cannot auto-detect cache config, using defaults.
--7387--      Run with -v to see.
n = 1025
timeSimple: 225.83
==7387==
==7387== I   refs:      50,410,471,110
==7387== I1  misses:           1,676
==7387== L1i misses:           1,658
==7387== I1  miss rate:           0.00%
==7387== L1i miss rate:          0.00%
==7387==
==7387== D   refs:      25,298,503,735 (23,994,547,215 rd + 1,303,956,520 wr)
==7387== D1  misses:      332,321,761 ( 332,114,921 rd +      206,840 wr)
==7387== L1d misses:      67,578,788 ( 67,380,472 rd +      198,316 wr)
==7387== D1  miss rate:           1.3% (           1.3% +           0.0% )
==7387== L1d miss rate:           0.2% (           0.2% +           0.0% )
==7387==
==7387== LL refs:      332,323,437 ( 332,116,597 rd +      206,840 wr)
==7387== LL misses:      67,580,446 ( 67,382,130 rd +      198,316 wr)
==7387== LL miss rate:           0.0% (           0.0% +           0.0% )
```

Настройки cachegrind

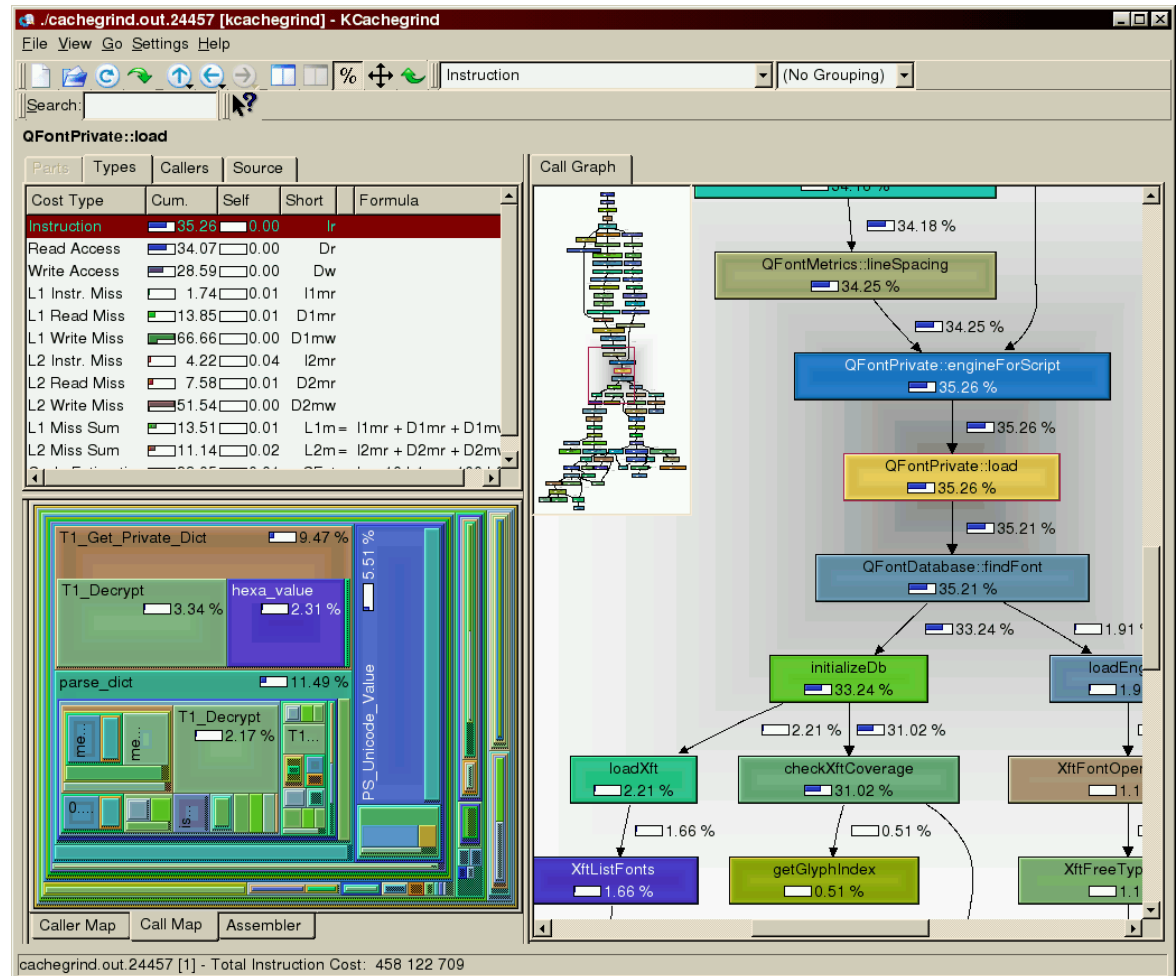
- Cache-simulation specific options are:
 - --I1=<size>,<associativity>,<line size> Specify the size, associativity and line size of the **level 1 instruction cache**.
 - --D1=<size>,<associativity>,<line size> Specify the size, associativity and line size of the **level 1 data cache**.
 - --L2=<size>,<associativity>,<line size> Specify the size, associativity and line size of the **level 2 cache**.

Параметры кэшей процессора (windows)

CPU	Caches	Mainboard	Memory	SPD	Graphics	Bench	About
L1 D-Cache							
Size	32 KBytes	x 2					
Descriptor	8-way set associative, 64-byte line size						
L1 I-Cache							
Size	32 KBytes	x 2					
Descriptor	8-way set associative, 64-byte line size						
L2 Cache							
Size	256 KBytes	x 2					
Descriptor	8-way set associative, 64-byte line size						
L3 Cache							
Size	4 MBytes						
Descriptor	16-way set associative, 64-byte line size						
Size							
Descriptor							
Speed							

Визуализация cachegrind

- kcachegrind



Собственный анализатор

- Определить параметры кэша на вашем ноутбуке
- Дополнить работу программы по вычислению произведения матриц анализом, в какую ячейку кэша попадает ваш запрос к данным
- Используя полученные данные, определить количество кэш промахов в модели:
 - [базовая оценка 15 баллов] вытеснения самой давно использованной кэш-линии
 - [+4 балла] вытеснения случайной кэш-линии (по желанию)

Форма отчета

- [1 балл] Результат определения параметров кэш памяти
- [1 балл] Параметры ноутбука (десктопа) для эксперимента
- [7 баллов] Код анализа количества кэш промахов
- [4 балла] Графики полученных количеств для различных n (обязательно 1023, 1024, 1025, 1040, 1041, 1050, 1100)
- [2 балла] Сравнение с количеством кэш промахов, предсказанным *valgrind --tool=cachegrind* (если есть возможность)

Дополнительное задание (по желанию)

- [+2 балла] При анализе кэш промахов можно собирать статистику, как давно была в кэш записана нужная линия памяти
 - Была записана k записей назад (считаем только записи в эту ячейку из памяти) и она еще в кэше
 - Была записана k записей назад и она уже вытеснена из кэша
- [+2 балла] Построить гистограммы полученных величин (должно получиться 2 или 4)
- [+2 балла] Написать код транспонирования матрицы, прогнать его через анализатор + valgrind

Спасибо за внимание!

- Презентация будет выложена в <http://anytask.org/course/70>