



LINUX PERF GUI (HOTSPOT)

**Профилировщик
с открытым исходным кодом
и функцией дизассемблера**

Дизассемблер

Новый программный компонент,
разработанный ПАО «НИПС»

Отсутствовал в исходном open-source коде

Дизассемблер - это компонент системы, который генерирует и отображает дизассемблированный код.

Другими словами, переводит машинный код на язык ассемблера.

**Функции дизассемблера,
ускоряющие оптимизацию работы приложения:**

Находит «горячие» участки, узкие места и распределение событий **внутри функций** пользовательского приложения.

Визуализирует информацию в виде текста и таблиц. Данные отображаются с **точностью до строки в коде** или инструкции в ассемблере.

Кроссплатформенность - работает с несколькими аппаратными платформами и операционными системам: ARMv7, ARMv8, x86_64.

Просмотр дизассемблированного кода.

Интуитивно понятный пользовательский интерфейс.

Дизассемблер

Новый программный компонент,
разработанный ПАО «НИПС»

Отсутствовал в исходном open-source коде

Вычисленные «горячие» участки

perf.l.fp.cache-misses.data - Hotspot

File Settings View Help

Summary Bottom Up Top Down Flame Graph Caller / Callee Disassembly

Search

Assembly

/home/vlat/samples/coremark/coremark-1.exe: file format elf64-x86-64

0000000000025d0 <matrix_mul_vect>: 100.00%

```
25d0: endbr64
25d4: test %edi,%edi
25d6: je 2633 <matrix_mul_vect+0x63>
25db: push %r12
25dd: mov %edi,%r11d
25e0: push %rbp
25e1: lea 0x4(%rsi,%rax,4),%r12
25e6: mov %rcx,%rbp
25e9: mov %rsi,%rcx
25ec: push %rbx
25ed: xor %ebx,%ebx
25ef: nop
25f0: mov %rbp,%r8
25f3: mov %ebx,%eax
25f5: xor %r9d,%r9d
25f8: nopl 0x0(%rax,%rax,1)
2600: mov %eax,%esi
2602: movswl (%r8),%r10d
2606: add $0x1,%eax
2609: add $0x2,%r8
260d: movswl (%rdx,%rsi,2),%rci
```

Time Line

Search

Event Source: cache-misses

Source

0s 2s 4s 6s 8s 10s 12s 14s 16s 18s 20s

CPUs

CPU #0

Processes

coremark...

corem...

perf.l.fp.cache-misses.data - Hotspot

File Settings View Help

Summary Bottom Up Top Down Flame Graph Caller / Callee Disassembly

Search

Symbol	Binary	cache-misses (self)	cache-misses (incl.)
??		79.2%	79.3%
core_bench_list	coremark-1.exe	3.86%	13.4%
matrix_test	coremark-1.exe	3.85%	6.71%
??	ld-2.30.so	3.07%	5.24%
cruc16	coremark-1.exe	2.39%	11.3%
_di_addr	libc-2.30.so	2.29%	2.29%
core_list_find	coremark-1.exe	1.09%	16.6%
matrix_mul_matrix_bitextract	coremark-1.exe	1.06%	11.4%
core_state_transition	coremark-1.exe	1.02%	9.84%
core_list_mergesort	coremark-1.exe	0.775%	2.09%
core_bench_state	coremark-1.exe	0.673%	6.92%
matrix_mul_matrix	coremark-1.exe	0.243%	4.68%
calc_func	coremark-1.exe	0.0945%	0.137%
cmp_idx	coremark-1.exe	0.0936%	1.08%
matrix_mul_vect	coremark-1.exe	0.0571%	0.260%
ipl_table	ipl_tables.ko	0.0567%	0.0567%
cruc12	coremark-1.exe	0.0423%	0.349%
cmp_complex	coremark-1.exe	0.0396%	0.099%

all Binary

Filter In By Symbol

Filter Out By Symbol

Filter Out

Reset Filter

Time Line

Search

Event Source: cache-misses

Source

0s 2s 4s 6s 8s 10s 12s 14s 16s 18s 20s

CPUs

CPU #0

Processes

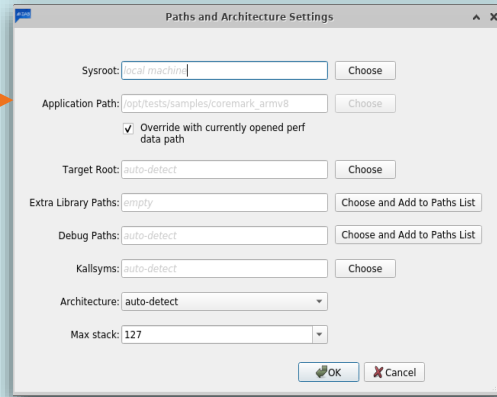
coremark...

corem...

Два подхода к генерации дизассемблера:
objdump и perf annotate, с возможностью их
переключения через UI.

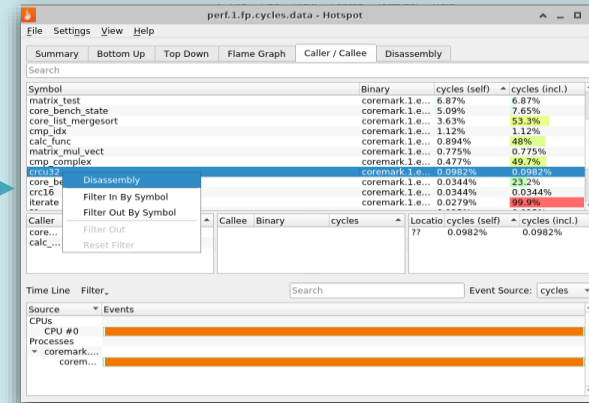
Варианты настроек Linux Perf GUI для анализа данных, собранных на машине с другой архитектурой

Удобный интерфейс, доработан ПАО «НИПС»



Операционная система Linux
Архитектура X86_64 (Host)

- Исполняемый файл
- perf.data

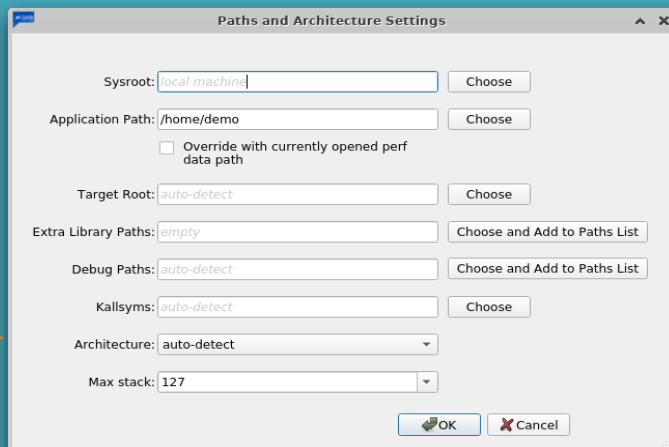


Операционная система Linux
Архитектура ARM (Target)

Анализ работы функций приложения

- Настройки вынесены в отдельное диалоговое окно.

- Выбор директории через File Browser.

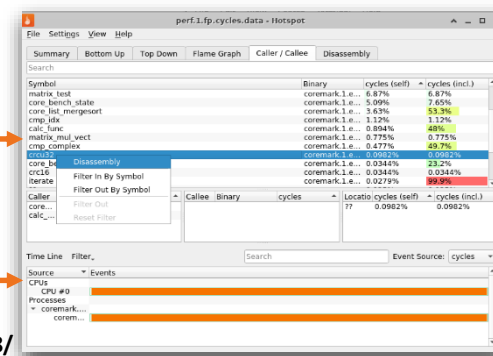


Операционная система Linux
Архитектура X86_64 (Host)

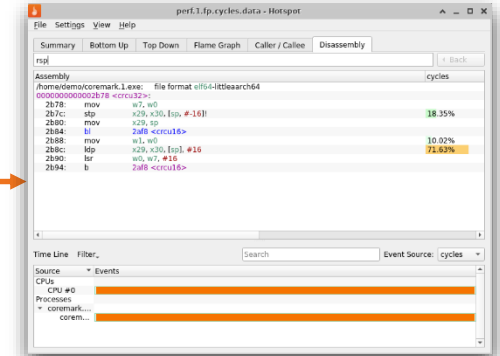
Исполняемый файл
/home/demo

perf.data
/home/demo/armv8/

Операционная система Linux
Архитектура ARM (Target)

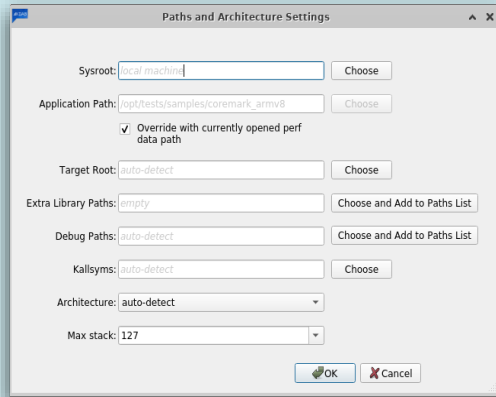


Анализ работы функций приложения



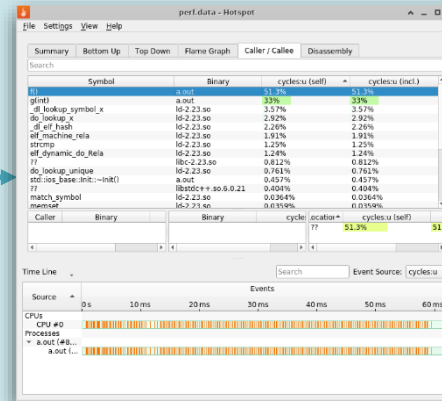
Анализ работы внутри функций приложения

Портирование Linux Perf GUI на платформу «Эльбрус» (доступно по запросу)



Операционная система Linux
Архитектура X86_64 (Host)

- Исполняемый файл
- perf.data



Архитектура E2K (Target)

Анализ работы
функций
приложения

В перспективе
портирование
Дизассемблера
на Эльбрус

```
mc [nips@elbrus:~]
Samples: 250 of event 'cycles:u', Event count (approx.): 40381591
Overhead Command Shared Object Symbol
51,29% a.out a.out [.] _Zlfv
32,95% a.out a.out [.] _Zlgi
3,57% a.out ld-2.23.so [.] _dl_lookup_symbol_x
2,92% a.out ld-2.23.so [.] do_lookup_x
2,26% a.out ld-2.23.so [.] _dl_elf_hash
1,91% a.out ld-2.23.so [.] elf_machine_rela
1,25% a.out ld-2.23.so [.] strcmp
1,24% a.out ld-2.23.so [.] elf_dynamic_do_Rela
0,76% a.out ld-2.23.so [.] do_lookup_unique
0,46% a.out a.out [.] 0x00000000000000a60
0,44% a.out libc-2.23.so [.] _dl_addr
0,40% a.out libstdc++.so.6.0.21 [.] _stt__14_locale_inst_cc_6523f73e
0,37% a.out libc-2.23.so [.] _wctype_l
0,04% a.out ld-2.23.so [.] match_symbol
0,04% a.out ld-2.23.so [.] memset
0,04% a.out ld-2.23.so [.] 0x00000000000000e88
0,04% a.out ld-2.23.so [.] _dl_map_object_deps
0,02% a.out ld-2.23.so [.] _dl_important_hwcaps
0,01% a.out ld-2.23.so [.] dl_main
0,01% a.out ld-2.23.so [.] dl_sysdep_start
0,00% a.out ld-2.23.so [.] dl_start
Tip: Customize output of perf script with: perf script -F event,ip,sym
```

perf report -i perf.data

Операционная система Эльбрус
Архитектура E2K

Преимущества **LINUX PERF GUI** **(HOTSPOT)**

Поддерживает архитектуры:
x86_64, ARMv7, ARMv8.

Вывод полученных данных в csv, текстовый формат.

Кросс Платформенность
Профилирование
Дизассемблирование

Высокая степень детализации данных о производительности, весах, временах: до строки внутри функции.

Поддерживает языки программирования **C, C++, Java, Fortran.**
Единственное требование - компиляция под Линукс.

Интуитивно понятный, многофункциональный интерфейс на английском языке.

Возможности развития Linux Perf GUI

ПЛАНЫ

Отображение исходного кода пользовательского приложения, соответствующего выводу дизассемблера:
Разработка включающих (inclusive) весов (costs) для дизассемблера по objdump-source.

НОВЫЕ ФУНКЦИИ ДИЗАССЕМБЛЕРА

Отображение исходного кода выбранной функции пользовательского приложения, соответствующего выводу дизассемблера – при наличии исходников. С возможностью их вывода совместно или порознь.

ПОДДЕРЖКА ДРУГИХ ОС, АРХИТЕКТУР

- Рассмотреть применимость AppImage для поддержки других ОС.
- Сборка на Windows, для работы в cross-режиме или, как альтернативу, изучить возможность применения AppImage.
- Поддержка Windows.
- Поддержка российской архитектуры «Эльбрус». Базовая функциональность портирована и доступна по запросу. Планируется портирование новой функциональности.

Возможности развития Linux Perf GUI

РАЗРАБОТКА ФУНКЦИОНАЛЬНОСТИ СРАВНИТЕЛЬНОГО АНАЛИЗА РАЗЛИЧНЫХ ВЕРСИЙ КОДА

РАЗРАБОТКА АНАЛИЗАТОРА ВЫДЕЛЕННЫХ ЧАСТЕЙ ПРИЛОЖЕНИЯ

Выделенные части пользовательского приложения (группы):

доступ к различным уровням памяти, операции с плавающей точкой, I/O, CPU/GPU взаимодействие и загрузка сетевых потоков и другие.

Подразумевает разбиение на группы и их распознавание. Распознавание может быть организовано на группировке семантически связанных системных вызовов POSIX стандарта. Учитывается доступная информация о событиях и их семантических связях.

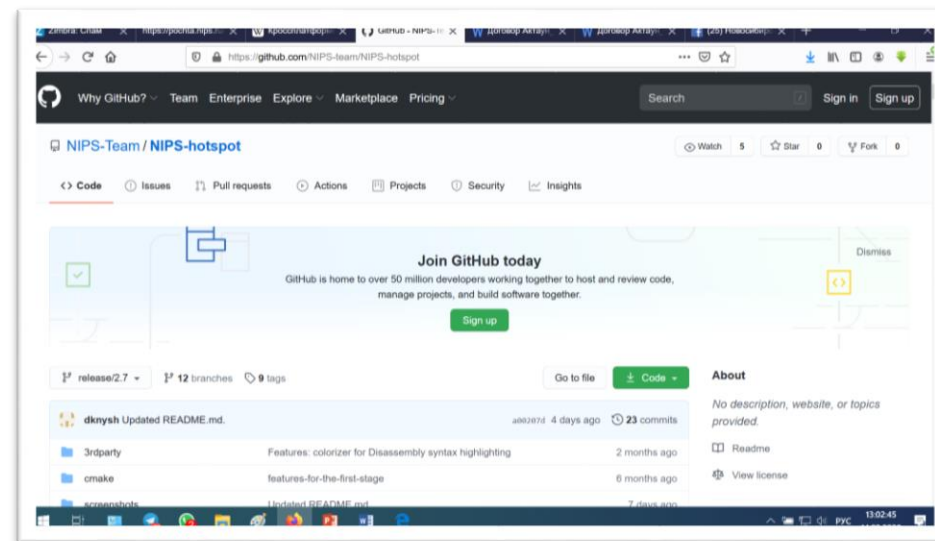
РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ ДЛЯ ГЕНЕРАЦИИ ОПТИМАЛЬНОГО КОДА

Может быть использовано для автоматического вычисления некоторых эвристик и результатов анализа, используемых компилятором. Например, поиск и предсказание наиболее вероятных весов инструкций для их генерации в наиболее подходящем порядке с целью минимизировать промахи для ветвлений.

Как работать с Linux Perf GUI

Вы загружаете к себе исходники из репозитория GitHub.
По приложенной [инструкции](#) устанавливаете и собираете систему.

Для работы с системой потребуются дополнительные компоненты,
такие как objdump. Они перечислены в [инструкции к установке](#).



Процесс профилирования



Техническая поддержка

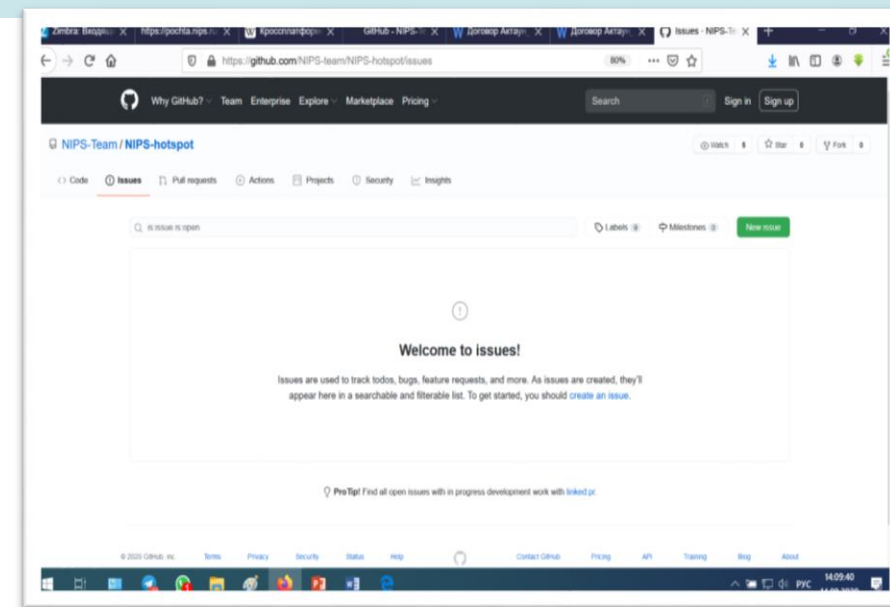
Linux Perf GUI — свободный бесплатный продукт.

На сайте продукта для скачивания всегда доступна стабильная версия. Сообщать о проблемах можно в [список запросов](#).

Наша команда оказывает помощь с построением, установкой и запуском продукта, устранением неполадок.

Разработка новой функциональности, крупных модулей системы, а также срочное решение проблем оказывается в рамках коммерческой поддержки.

Кроме того, на коммерческой основе предлагаем услуги по анализу и оптимизации кода и рефакторингу пользовательских приложений.



Коммерческая техническая поддержка

Коммерческая поддержка включает кастомизацию, разработку новой функциональности, оперативное устранение неполадок, регулярную поддержку.

В рамках коммерческой поддержки мы предоставляем готовые бинарные файлы, изменяем интерфейс и оптимизируем функционал по согласованному ТЗ.

Что происходит с кодом, написанным в рамках коммерческой поддержки?

Основная версия продукта и его обновления выкладываются в открытые репозитории [НИПС](#) и [KDAB](#).

Для коммерческих проектов заводятся частные репозитории, где хранятся изменения программного кода.

Они открываются для всеобщего доступа только по согласованию с заказчиком.

Процесс коммерческой поддержки

Заказ начинается с обсуждения потребностей клиента и составления ТЗ.

Подписывается договор и начинается разработка. Разрабатываемый код заливается в закрытый репозиторий и все взаимодействие с заказчиком происходит через него.

Регулирование программной разработки

LINUX PERF GUI развивается под эгидой международного фонда открытого ПО программного обеспечения Github ([KDAB](#)) и подчиняется всем его правилам.

LICENSE

Разработка и продвижение Hotspot регулируется двумя лицензиями: GPL v2+, коммерческой лицензией KDAB.



По вопросам доработки и поддержки Linux Perf GUI (Hotspot)



Олег Молородов

Руководитель
по инновационному
развитию ПАО «НИПС»

Тел/ Whatsapp:
[8-913-472-68-98](tel:8-913-472-68-98)

По техническим вопросам Дарья Кныш

Ведущий разработчик
и технический лидер проекта
Linux Perf GUI ПАО «НИПС»

E-mail:
d.knysh@nips.ru