

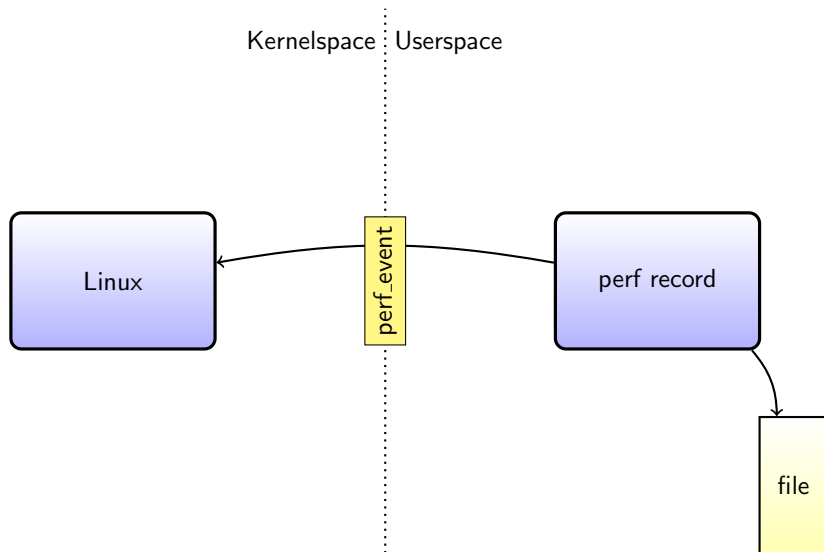
perf file format

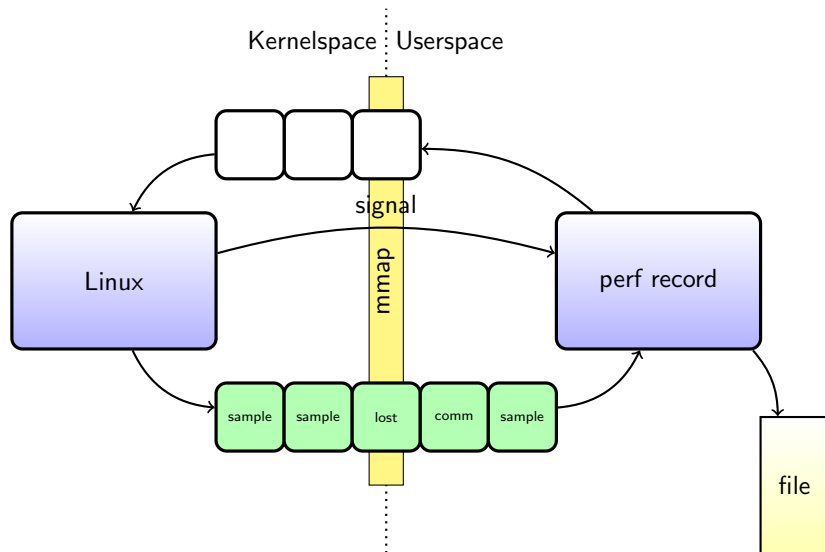
Urs Fässler

CERN Openlab

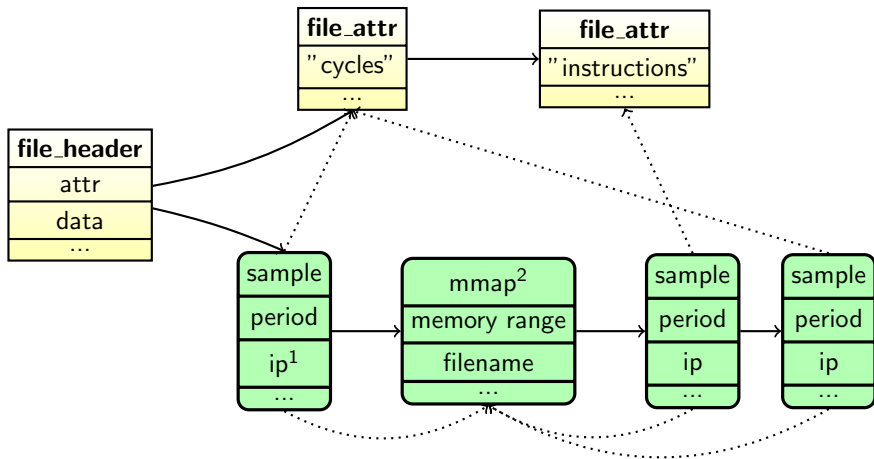
02.08.2011

initialization





perf file format (simplified, see page 6)



¹instruction pointer

²not the same as on the previous slide

Events: 1K cycles

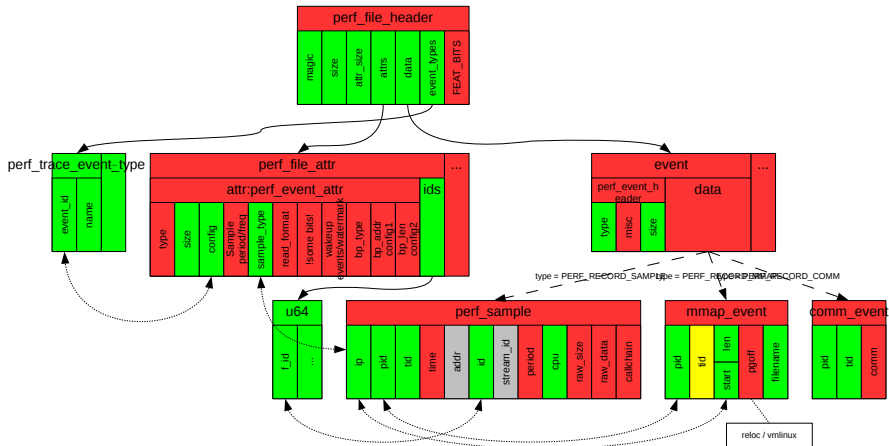
65.31%	1284	test	test	[.]	recFib
23.45%	461	test	test	[.]	loopFib
2.99%	59	test	libc-2.13.so	[.]	_int_malloc
1.61%	32	test	test	[.]	makeNode
1.48%	29	test	[kernel.kallsyms]	[k]	clear_page_c
1.11%	22	test	test	[.]	randTree
1.07%	21	test	libc-2.13.so	[.]	__malloc
0.56%	11	test	libc-2.13.so	[.]	random_r
0.50%	10	test	libc-2.13.so	[.]	random
0.35%	7	test	libc-2.13.so	[.]	@plt
0.21%	23.45	90549430	461	loopFib	_fault
0.20%	65.31	252203567	1284	recFib	free_one_page
0.15%	1.11	4283895	22	randTree	page_fault
0.10%	1.61	6218308	32	makeNode	_event_mmap
0.10%	5.66	21861152	111	??	alloc_pages_nodemask
0.10%	2.87	11076171	64		alloc_pages_vma
0.10%	2	test	[kernel.kallsyms]	[k]	up_vmas
0.06%	1	test	ld-2.13.so	[.]	get_page_from_freelist
0.06%	1	test	ld-2.13.so	[.]	_dl_map_object
0.06%	1	test	ld-2.13.so	[.]	_dl_relocate_object
0.06%	1	test	ld-2.13.so	[.]	bsearch
0.05%	1	test	[kernel.kallsyms]	[k]	debug_deactivate
0.05%	1	test	[kernel.kallsyms]	[k]	handle_pte_fault
0.05%	1	test	[kernel.kallsyms]	[k]	__rmqueue

Statistics per function:

per	period	samples	function
23.45	90549430	461	loopFib
65.31	252203567	1284	recFib
1.11	4283895	22	randTree
1.61	6218308	32	makeNode
5.66	21861152	111	??
2.87	11076171	64	

Press '?' for help on key bindings

detailed file format



map ip to function name

- ① sample contains Instruction Pointer
- ② find corresponding mmap event³
 - mmap events contains binary name
- ③ addr2line shows source file and function name from IP and binary
- ④ aggregate all samples in the same function

³IP has to be in range of mmap

- understand timestamp of sample
- mapping instruction pointer to library functions
- understand how the address \leftrightarrow filename works

- good understanding of file format
 - map samples to functions
 - same result as perf report
 - library and system function doesn't work entirely

Analyze virtualized machine

`perf` has support for the Kernel-based Virtual Machine (KVM [1]). For the performance measurement, an argument tells `perf` that the machine using KVM should be monitored. It uses the PMU of the host. It seems that also detailed information is available if the host machine has access to the guests `/proc/` files. Measures of hardware counters from inside the virtual machine is not supported.

VirtualBox [2] is not supported. This has two consequences. First, `perf` on the host can not record data about the guest. Second, there is no PMU in the virtual machine and therefore `perf` can not record hardware counters.

- [1] *Kernel-based Virtual Machine*. URL:
http://www.linux-kvm.org/page/Main_Page.
- [2] *VirtualBox*. URL: <http://www.virtualbox.org/>.