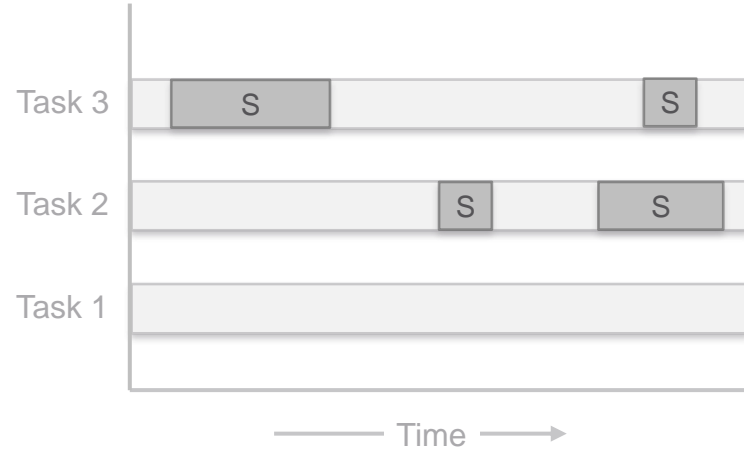




Improve Linux Perf tool to account for task sleep

Improve Linux Perf tool to account for task sleep



Agenda

- Problem Statement: Linux Perf tool not monitoring sleeping tasks
- Perf tool architecture:
 - Sample collection
 - No sample when target is sleeping
- Solutions:
 - Calculate sleep time and add to total sample count
 - Capture sleep sample

Problem Statement: Linux Perf tool not monitoring sleeping task

Perf tool doesn't collect samples if target process is in sleep state which leads to:

- Incorrect 'CPU usage' calculation:
If target task was in sleep state for around 50% of the time, the CPU usage represented by the perf tool does not account for the same.
- No 'task sleep time':
As the perf tool does not provide any sleep sample, it is not possible to determine for how long the task was in sleep state.



Function 1 has sample 1, 6 = 2 samples
Function 2 has sample 2, 3, 4, 5, 7, 8 = 6 samples
Total samples = 8

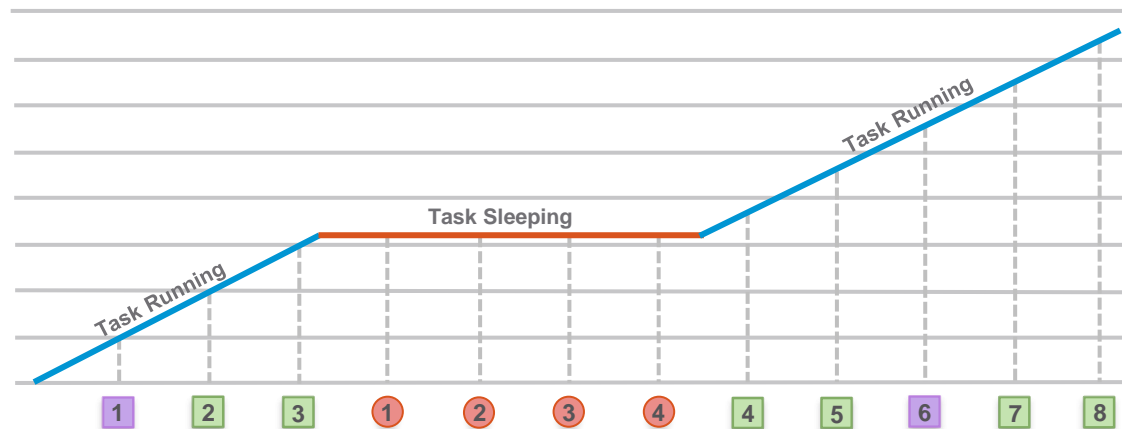
CPU usage calculation:

Function 1 CPU usages = $(2/8) * 100 = 25\%$
Function 2 CPU usages = $(6/8) * 100 = 75\%$

Problem Statement: Linux Perf tool not monitoring sleeping task

Perf tool doesn't collect samples if target process is in sleep state which leads to:

- Incorrect 'CPU usage' calculation:
If target task was in sleep state for around 50% of the time, the CPU usage represented by the perf tool does not account for the same.
- No 'task sleep time':
As the perf tool does not provide any sleep sample, it is not possible to determine for how long the task was in sleep state.

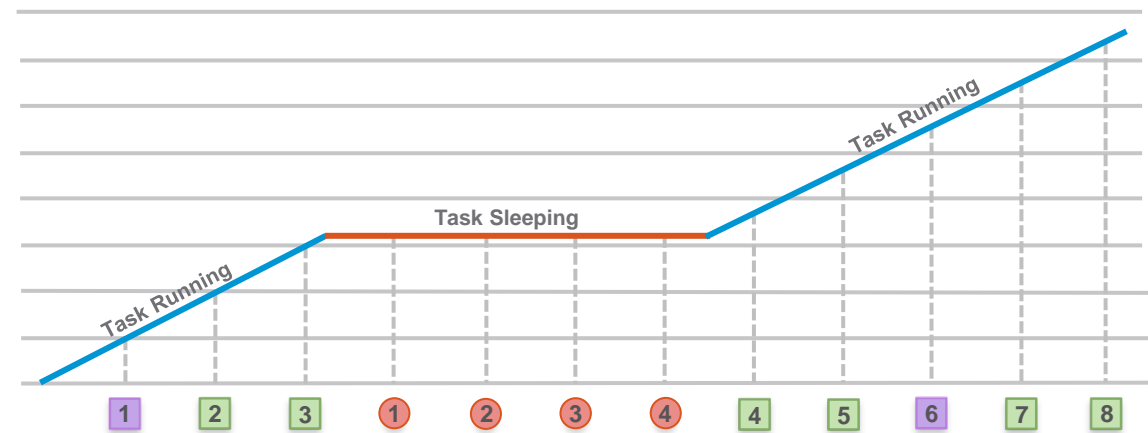


Function 1 has sample 1, 6 = 2 samples
Function 2 has sample 2, 3, 4, 5, 7, 8 = 6 samples
Total samples = 8

CPU usage calculation:

Function 1 CPU usages = $(2/8) * 100 = 25\%$
Function 2 CPU usages = $(6/8) * 100 = 75\%$

Problem Statement: Linux Perf tool not monitoring sleeping task



- Function 1 has sample 1, 6 = 2 samples
 - Function 2 has sample 2, 3, 4, 5, 7, 8 = 6 samples
- Total samples = 8

CPU usage calculation:

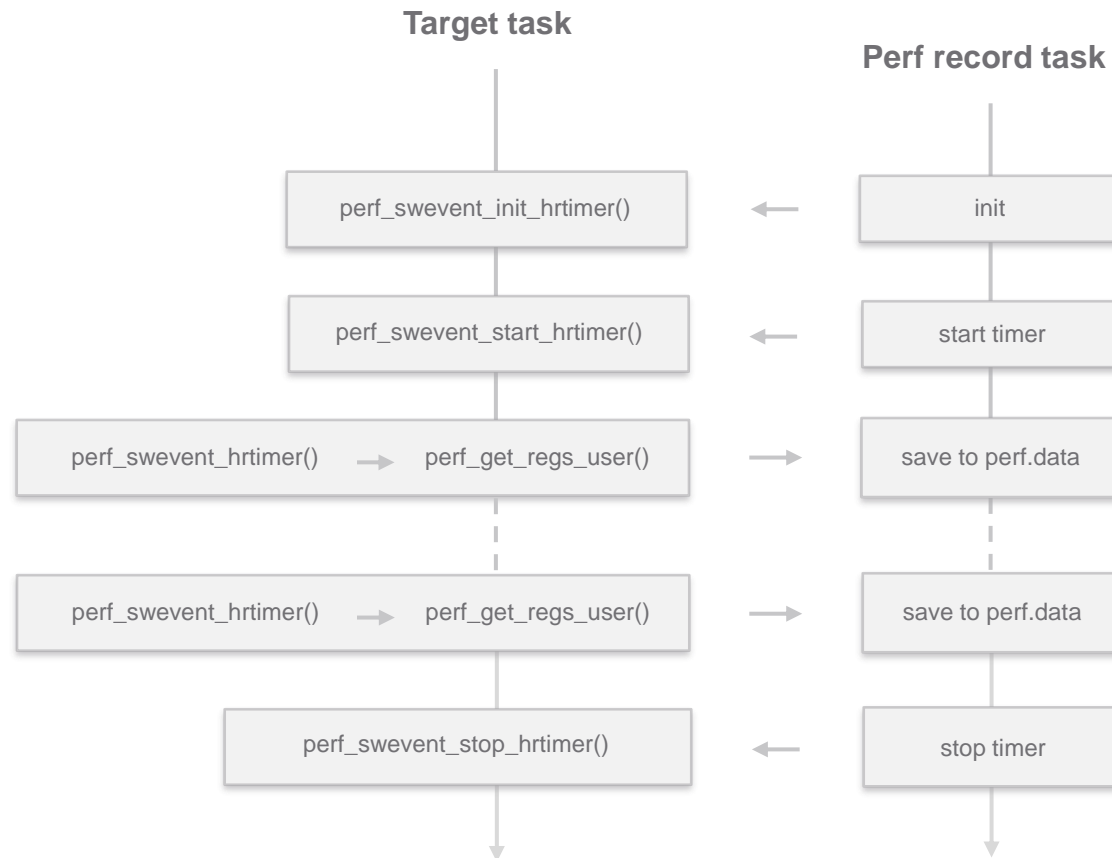
- Function 1 CPU usages = $(2/8) \times 100 = 25\%$
- Function 2 CPU usages = $(6/8) \times 100 = 75\%$

- Function 1 has sample 1, 6 = 2 samples
 - Function 2 has sample 2, 3, 4, 5, 7, 8 = 6 samples
- Total samples = 8 + 4 (skipped samples) = 12

CPU usage calculation:

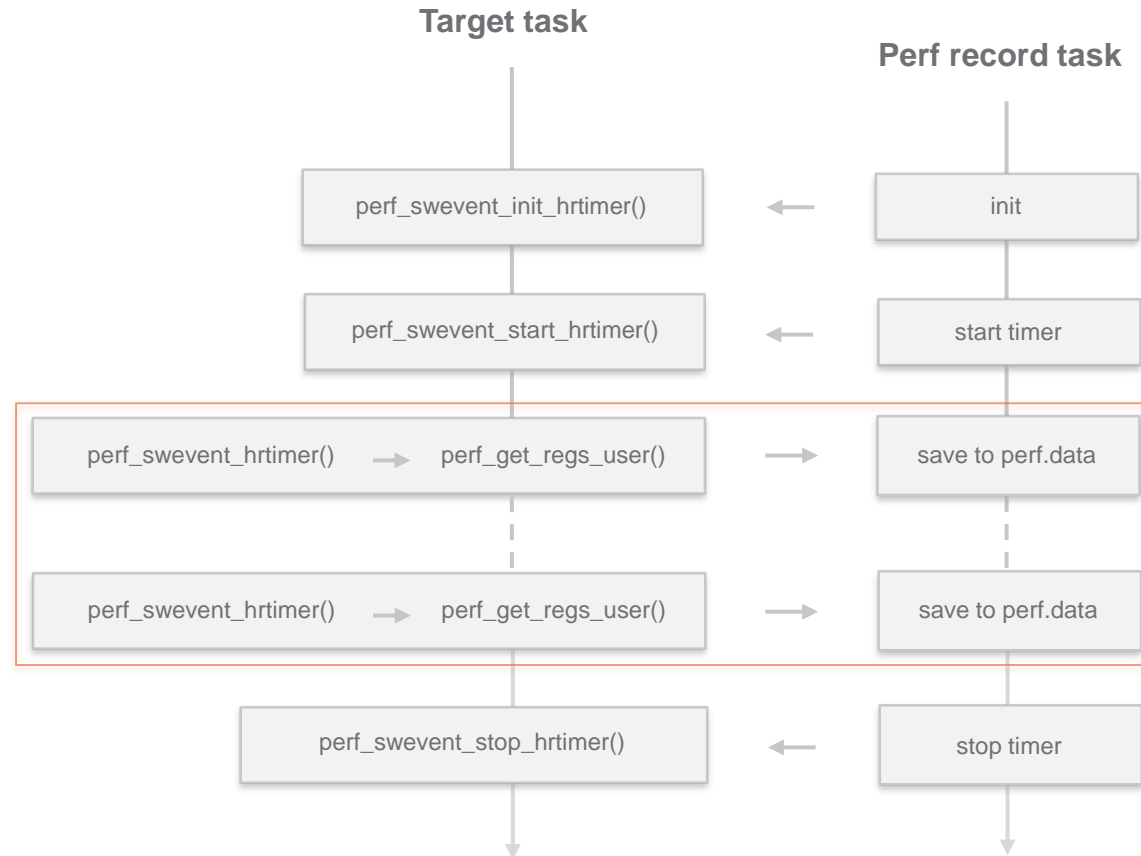
- Function 1 CPU usages = $(2/12) \times 100 = 16.6\%$
 - Function 2 CPU usages = $(6/12) \times 100 = 50\%$
- Sleep time = $(4/12) \times 100 = 33.3\%$

Perf tool architecture: Sample collection



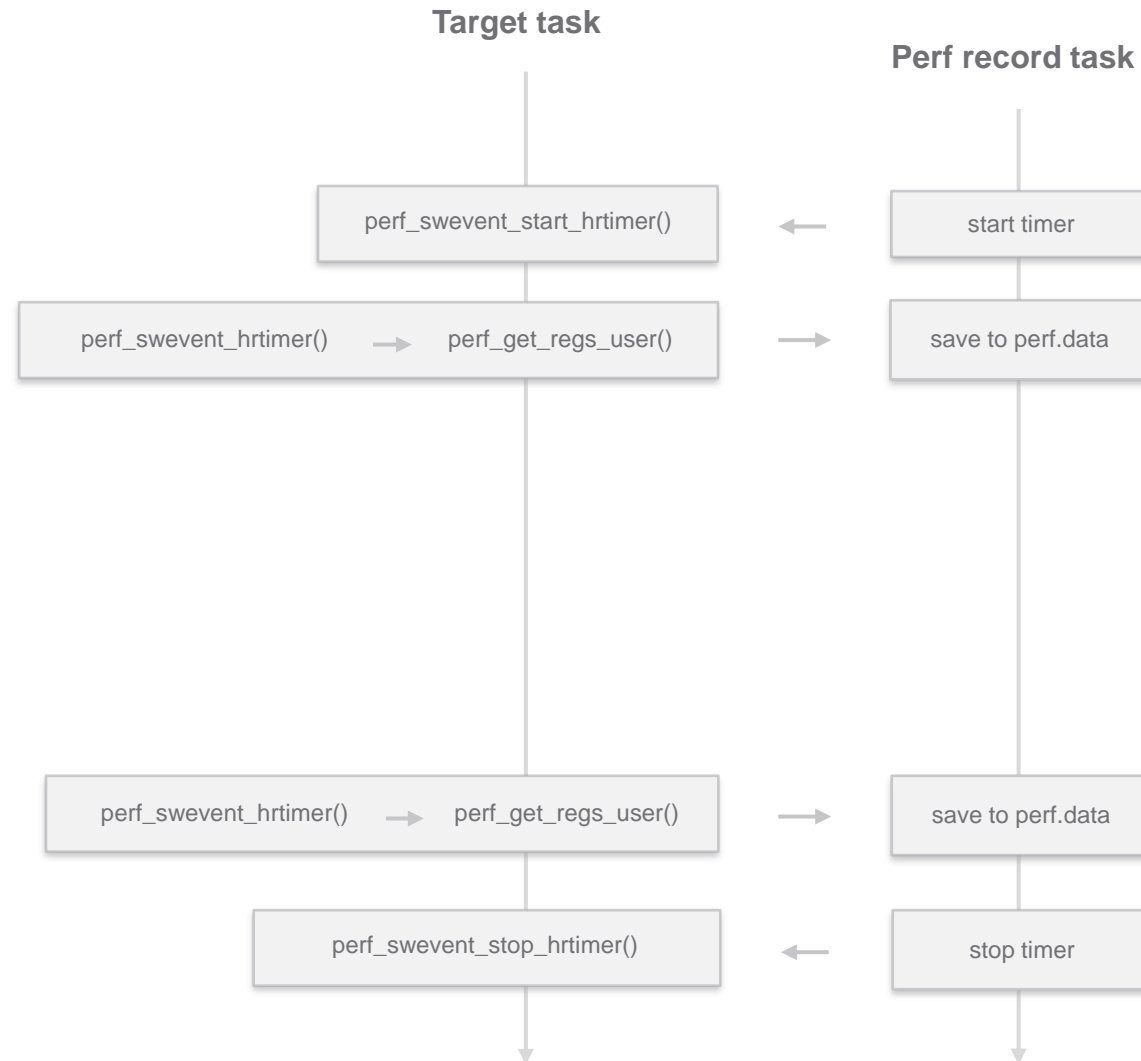
Execution flow: Perf collecting samples

Perf tool architecture: Sample collection

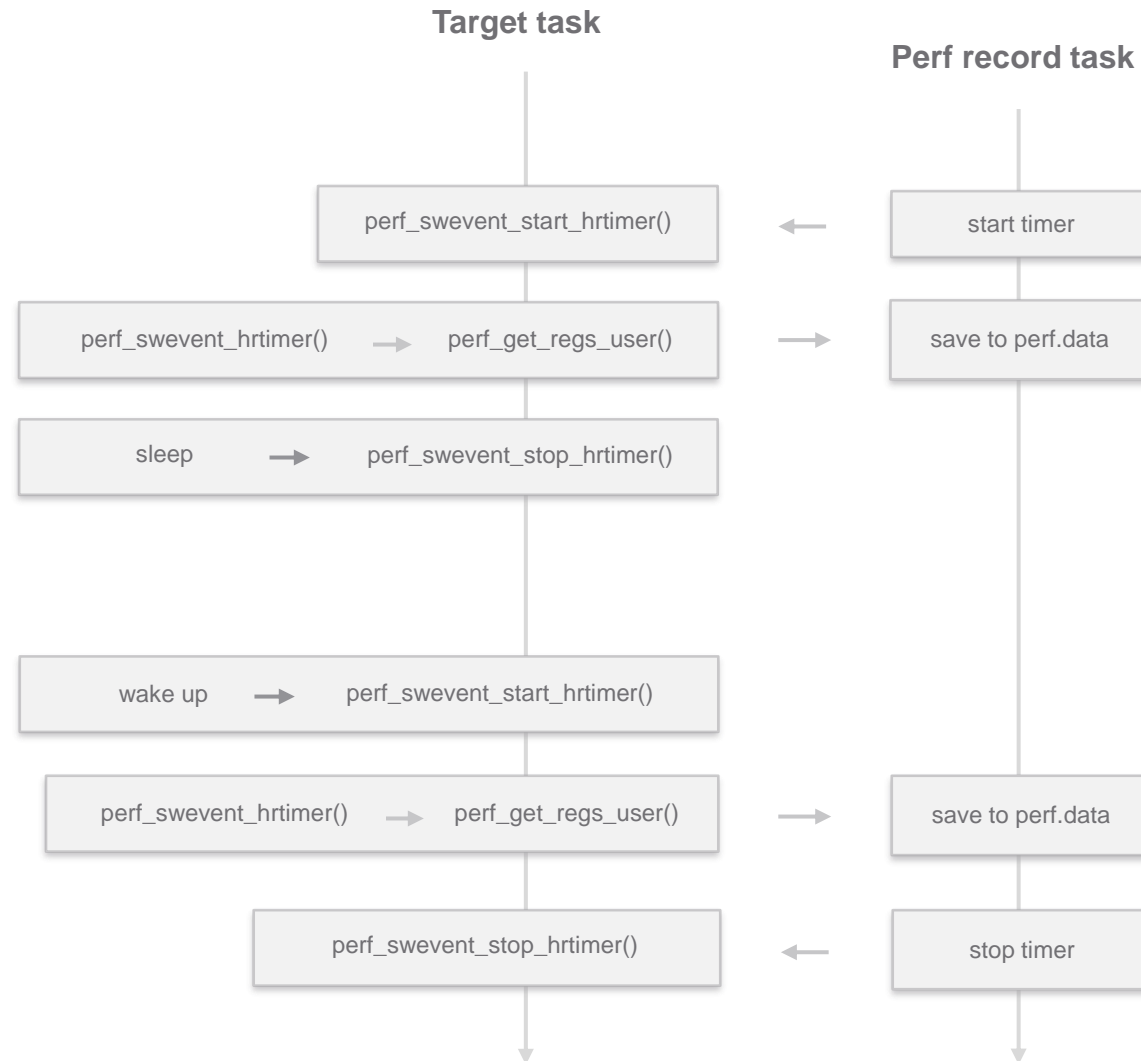


Execution flow: Perf collecting samples

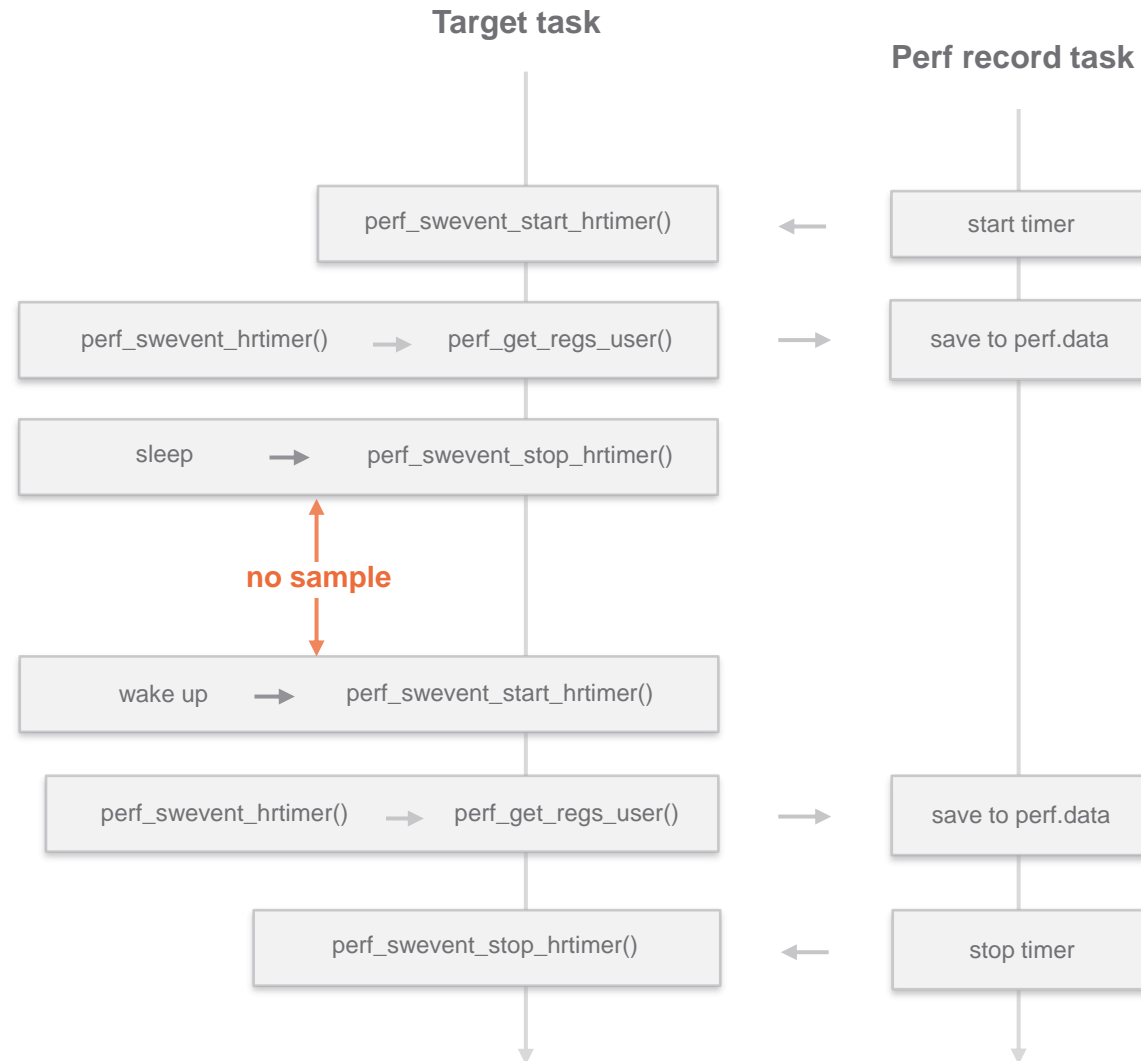
Perf tool architecture: No sample when target is in sleep



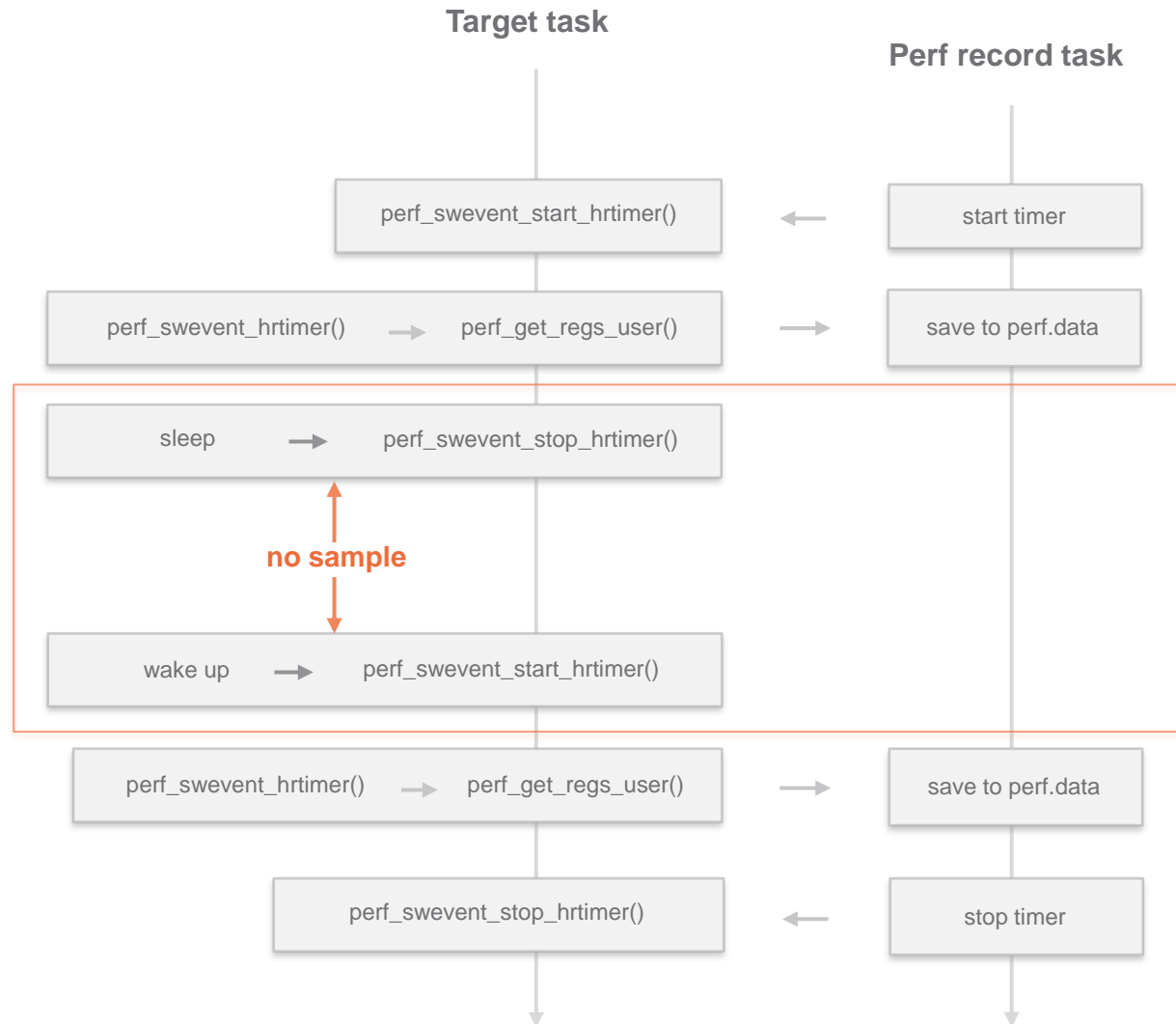
Perf tool architecture: No sample when target is in sleep



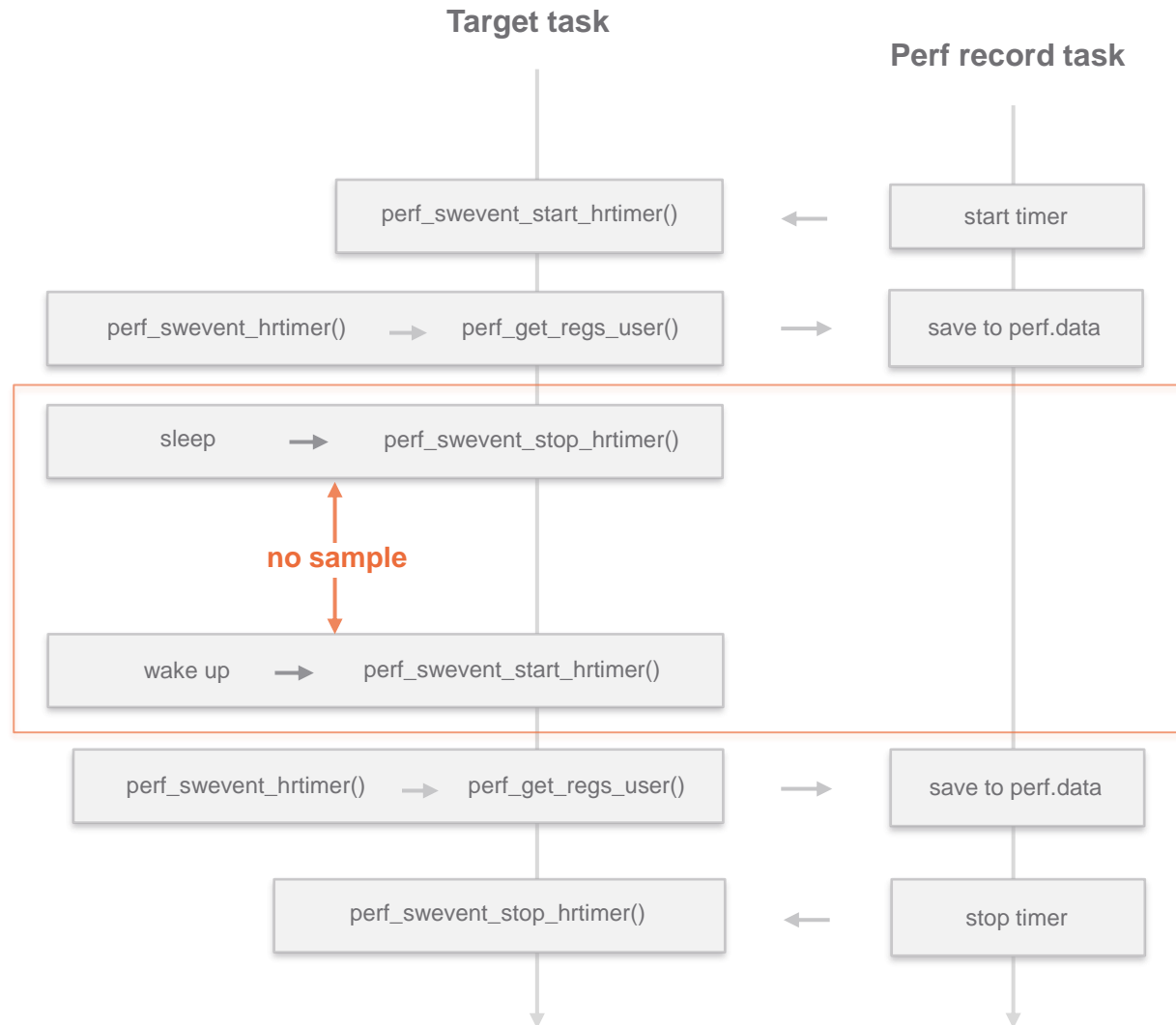
Perf tool architecture: No sample when target is in sleep



Perf tool architecture: No sample when target is in sleep

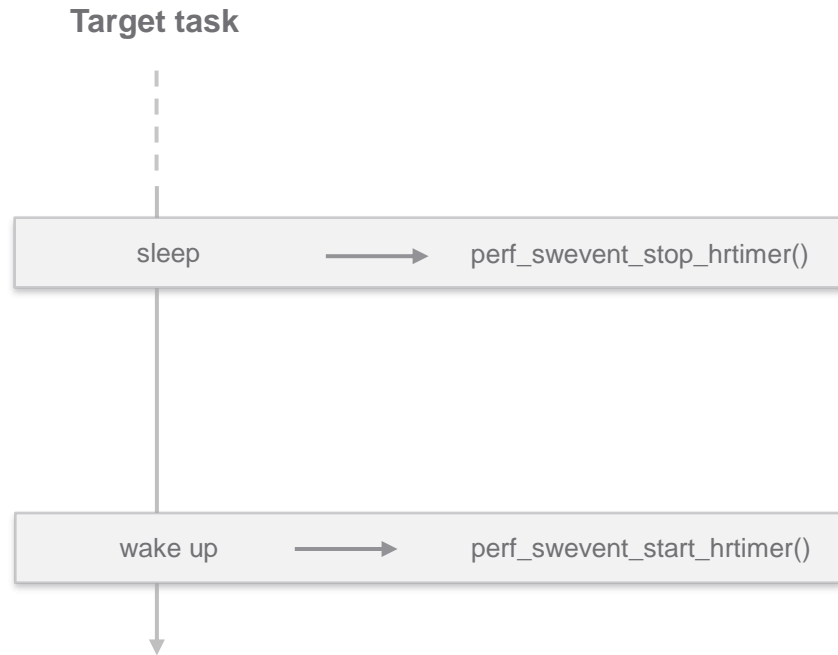


Perf tool architecture: No sample when target is in sleep

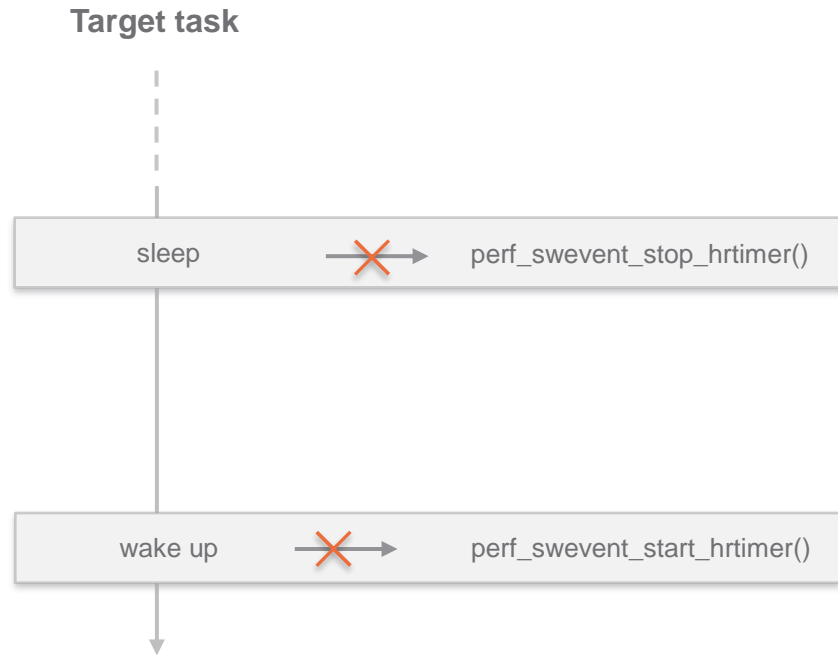


Solution #1: Capture sleep sample

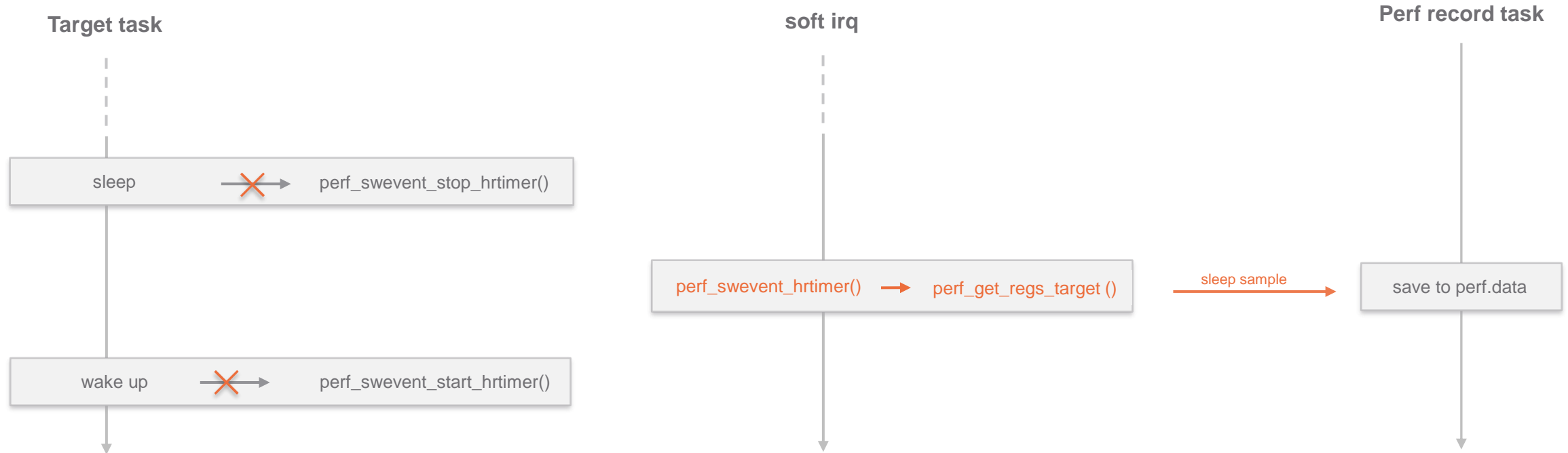
Solution #1: Capture sleep sample



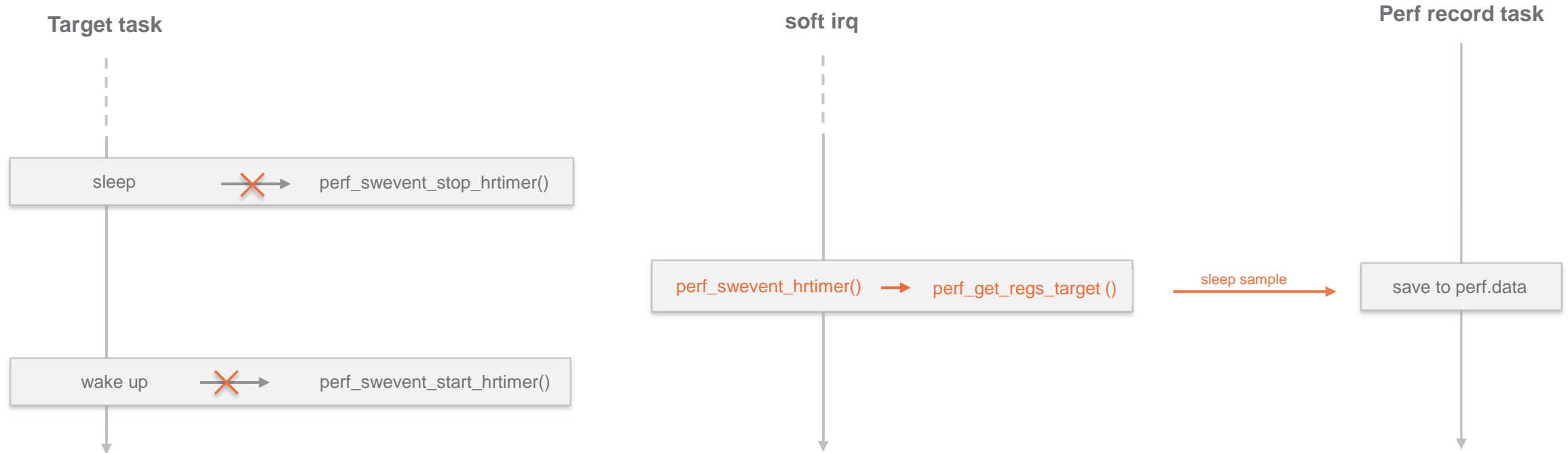
Solution #1: Capture sleep sample



Solution #1: Capture sleep sample



Solution #1: Capture sleep sample

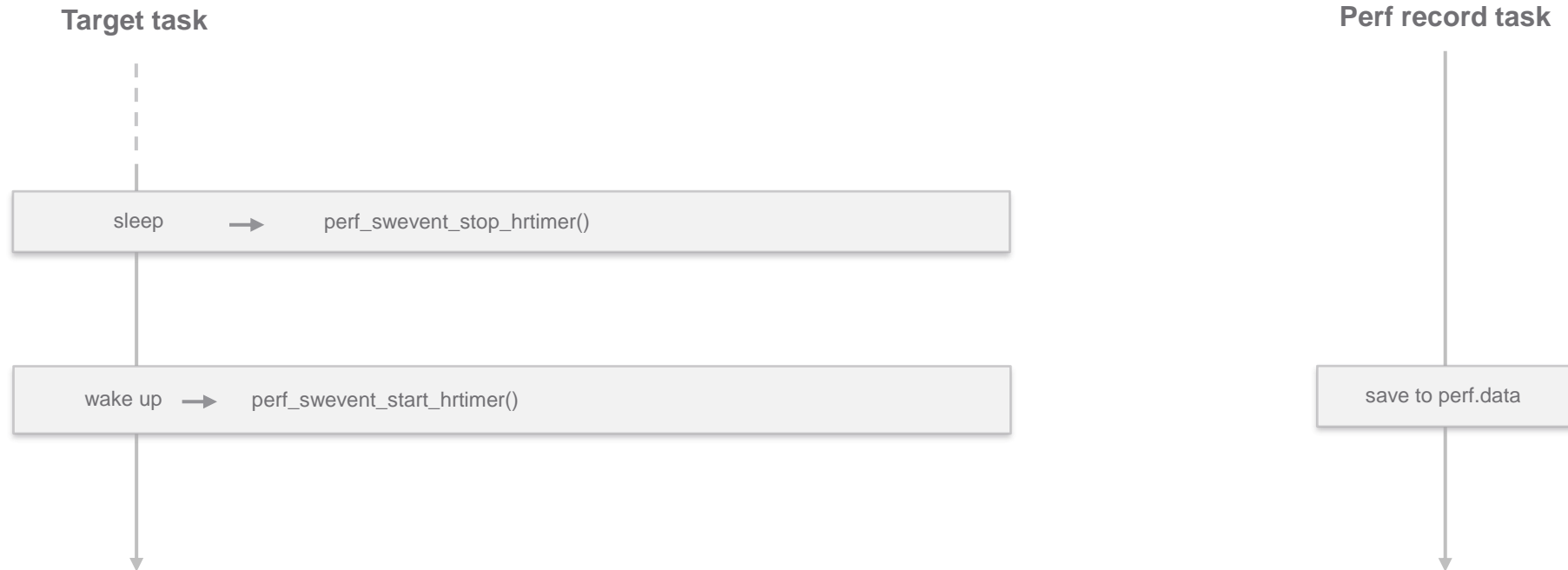


Disadvantages: size of perf.data will increase.

Solution

Solution #2: Calculate sleep time and add to total sample count

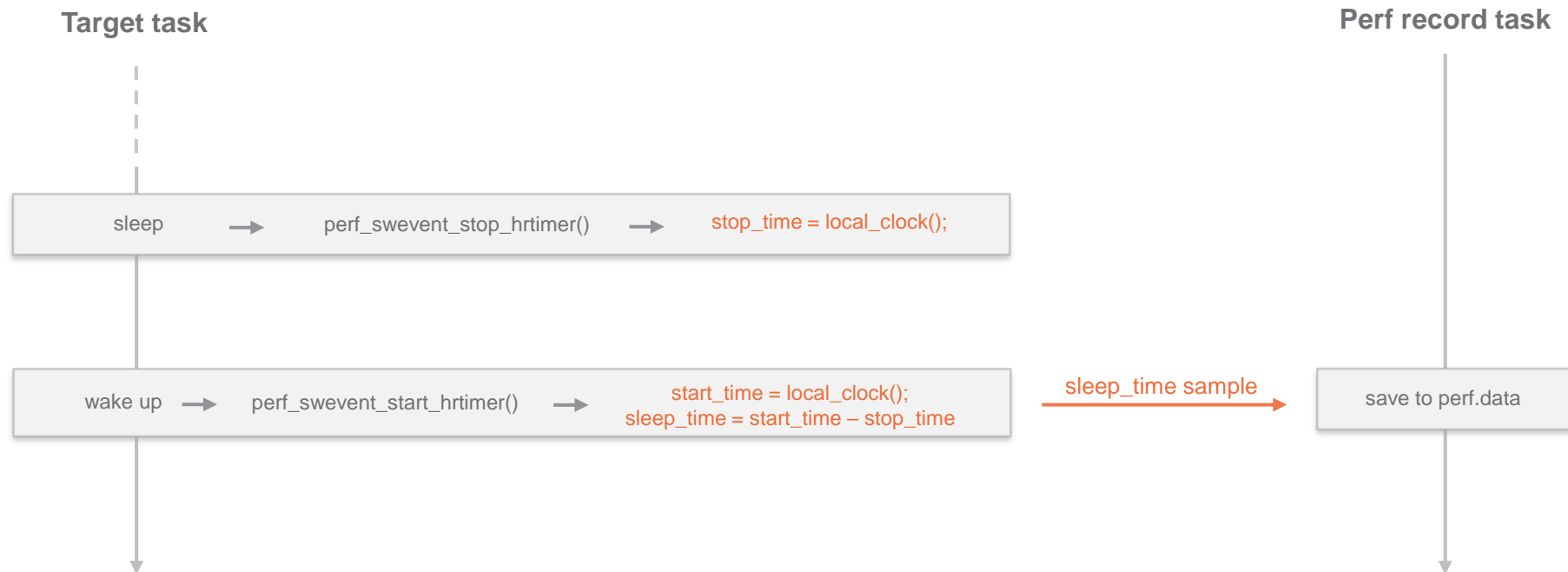
Solution #2: Calculate sleep time and add to total sample count



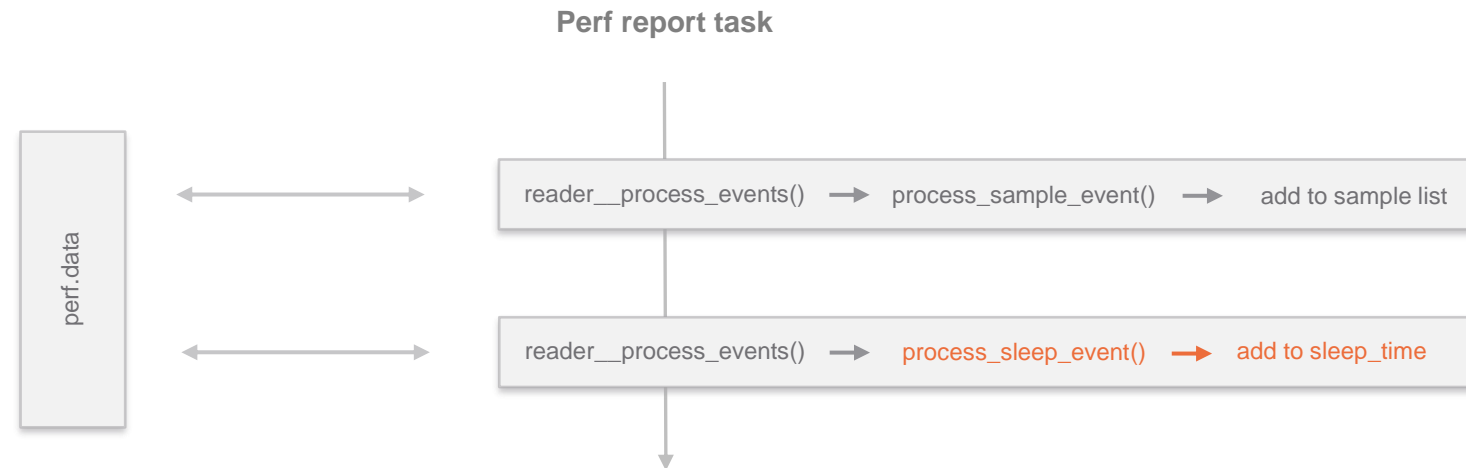
Solution #2: Calculate sleep time and add to total sample count



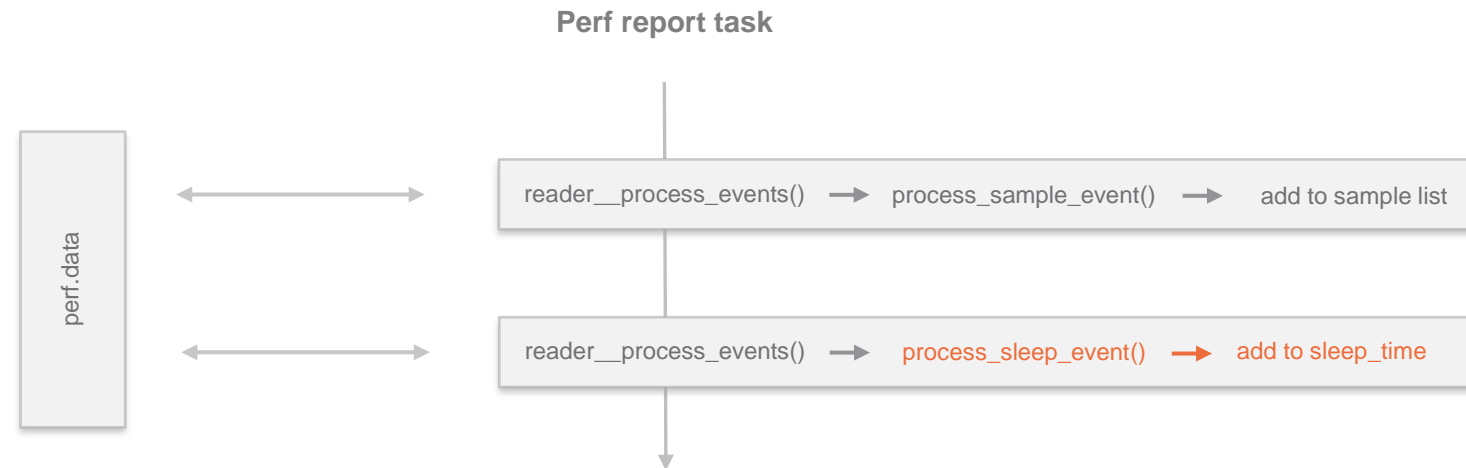
Solution #2: Calculate sleep time and add to total sample count



Solution #2: calculate sleep time and add to total sample count



Solution #2: calculate sleep time and add to total sample count

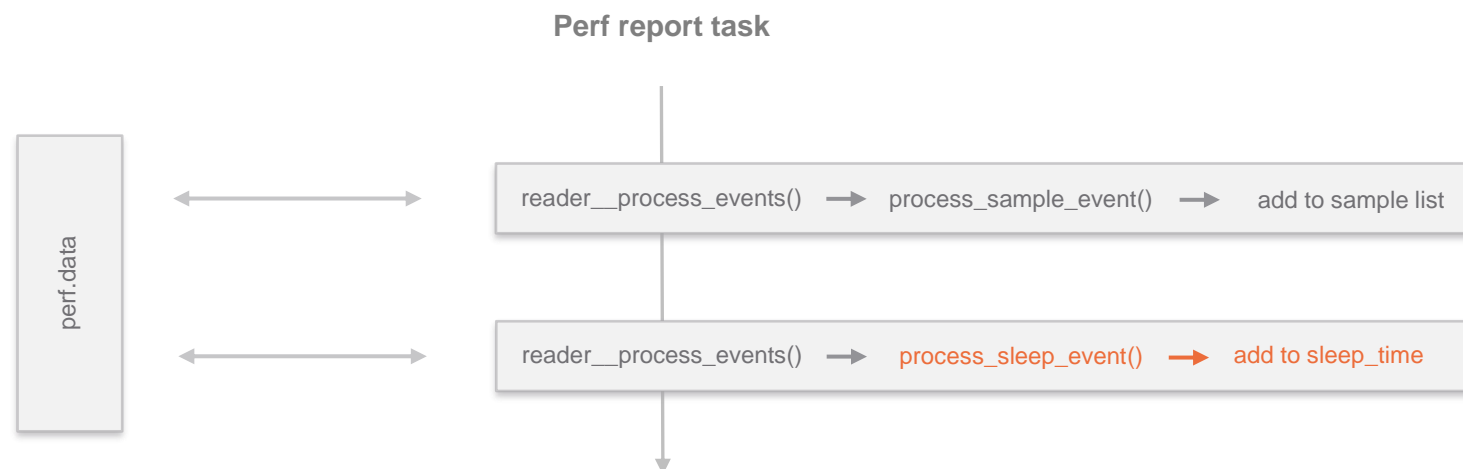


$\text{sample_time} = 1 \text{ second} / \text{sample_per_second}$

$\text{samples_time} = \text{sample_count} * \text{sample_time}$

$\text{total_time} = \text{samples_time} + \text{sleep_time}$

Solution #2: calculate sleep time and add to total sample count



$\text{sample_time} = 1 \text{ second} / \text{sample_per_second}$

$\text{samples_time} = \text{sample_count} * \text{sample_time}$

$\text{total_time} = \text{samples_time} + \text{sleep_time}$

$\text{CPU usage \%} = (\text{samples_time} / \text{total_time}) * 100$

$\text{sleep \%} = (\text{sleep_time} / \text{total_time}) * 100$

Perf tool snapshots:

```
int main(int argc, char* argv[]) {
    int pwm = strtol(argv[1], NULL, 0);

    while (1) {
        struct timeval stop, c;
        gettimeofday(&stop, NULL);
        stop.tv_sec++;
        usleep((100 - pwm) * 10000);
        while (1) {
            gettimeofday(&c, NULL);
            if (c.tv_sec > stop.tv_sec)
                break;
            if (c.tv_sec == stop.tv_sec && c.tv_usec > stop.tv_usec)
                break;
        }
        return 0;
    }
}
```

./a.out 10 (90% sleep, 10% duty cycle)

Samples: 4 of event 'cpu-clock:ppp', Event count (approx.): 400000000					
Children	Self	Command	Shared Object	Symbol	
- 100.00%	100.00%	a.out	[vdso]	[.] __vdso_gettimeofday	
_start					
__libc_start_main					
0x7f7fe0ea5189					
main					
__vdso_gettimeofday					
+ 100.00%	0.00%	a.out	a.out	[.] _start	
+ 100.00%	0.00%	a.out	libc.so.6	[.] __libc_start_main	
+ 100.00%	0.00%	a.out	libc.so.6	[.] 0x00007f7fe0ea5189	
+ 100.00%	0.00%	a.out	a.out	[.] main	

perf record --call-graph dwarf -p `pidof a.out` -F 10 -- sleep 4

Total expected samples = 40
Actual sample collected = 4 (which is 10% of 40)

Perf tool snapshots:

Samples: 4 of event 'cpu-clock:ppp', Event count (approx.): 400000000

Children	Self	Command	Shared Object	Symbol
- 100.00%	100.00%	a.out	[vdso]	[.] __vdso_gettimeofday
				_start
				__libc_start_main
				0x7f7fe0ea5189
				main
				__vdso_gettimeofday
+ 100.00%	0.00%	a.out	a.out	[.] _start
+ 100.00%	0.00%	a.out	libc.so.6	[.] __libc_start_main
+ 100.00%	0.00%	a.out	libc.so.6	[.] 0x00007f7fe0ea5189
+ 100.00%	0.00%	a.out	a.out	[.] main

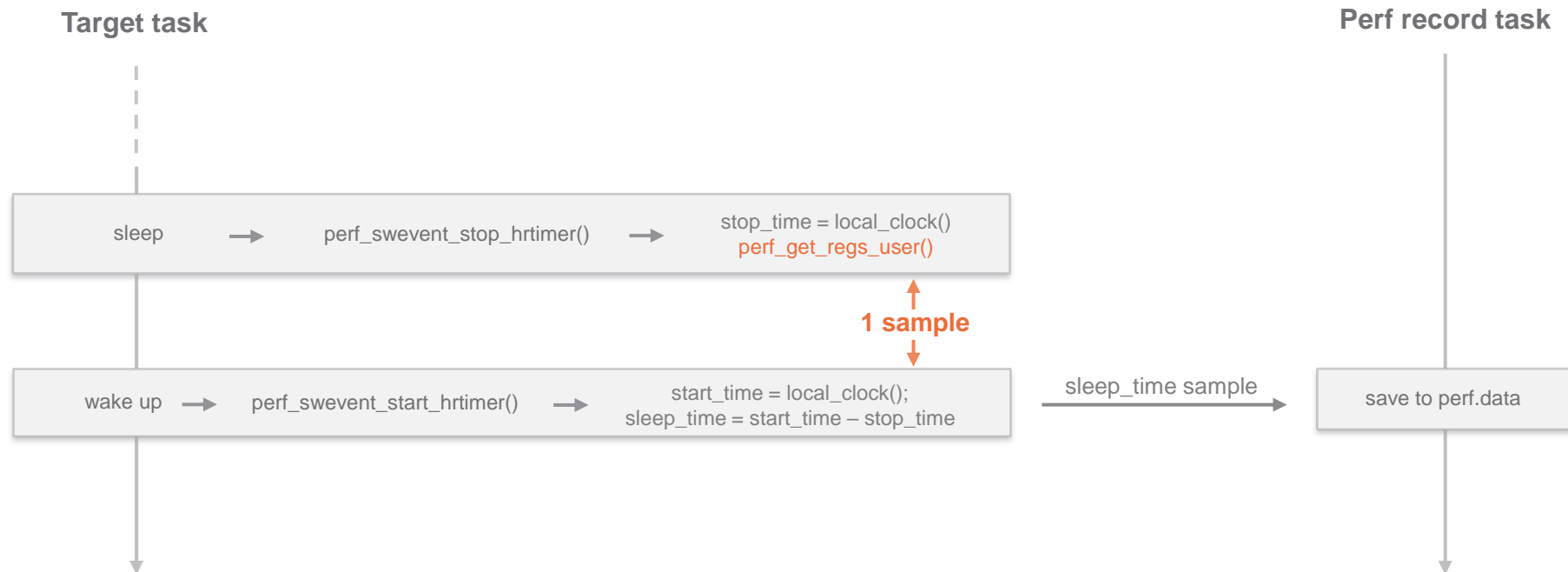
Existing Perf tool

Samples: 4 (10%), sleep samples: 36 (90%) of event 'cpu-clock:ppp', Event count (approx.): 4000000000

Children	Self	Command	Shared Objects	Symbol
- 10.00%	10.00%	a.out	[vdso]	[.] __vdso_gettimeofday
				_start
				__libc_start_main
				0x7f7fe0ea5189
				main
				__vdso_gettimeofday
+ 10.00%	0.00%	a.out	a.out	[.] _start
+ 10.00%	0.00%	a.out	libc.so.6	[.] __libc_start_main
+ 10.00%	0.00%	a.out	libc.so.6	[.] 0x00007f7fe0ea5189
+ 10.00%	0.00%	a.out	a.out	[.] main

Solution #2: Perf tool

Solution #2: Calculate sleep time and add to total sample count





HAVE ANY
QUESTIONS AND DOUBTS



Thanks