



profiling scripts with perf

jiri olsa

PROFILING SCRIPTS

- **profile interpreter**
VS
- **profile script**



PROFILING SCRIPTS

- profile interpreter

```
$ perf record python script.py
```

```
$ perf report
```

#	Overhead	Command	Shared Object	Symbol
#
#				
	9.71%	yum	libpython3.6m.so.1.0	[.] _PyEval_EvalFrameDefault
	5.08%	yum	libpython3.6m.so.1.0	[.] _init
	3.00%	yum	libpython3.6m.so.1.0	[.] _PyType_Lookup
	2.44%	yum	libpython3.6m.so.1.0	[.] PyObject_Malloc
	2.06%	yum	ld-2.25.so	[.] do_lookup_x
	1.88%	yum	libpython3.6m.so.1.0	[.] PyDict_GetItem
	1.57%	yum	libpython3.6m.so.1.0	[.] _PyObject_GenericGetAttrWithDict
	1.30%	yum	ld-2.25.so	[.] strcmp
	1.29%	yum	libpython3.6m.so.1.0	[.] _Py_HashBytes
	1.22%	yum	libpython3.6m.so.1.0	[.] PyDict_SetDefault
	1.13%	yum	libpython3.6m.so.1.0	[.] 0x00000000001153e0
	1.05%	yum	libpython3.6m.so.1.0	[.] _PyDict_LoadGlobal



PROFILING SCRIPTS

- profile script

```
$ perf record python script.py
```

```
$ perf report
```

#	Overhead	Command	Script File	Symbol
#
	23.67%	python3	/usr/lib64/python3.6/site-packages/hawkey/__init__.py	[.] run+0x0
	23.55%	python3	/usr/lib/python3.6/site-packages/dnf/cli/output.py	[.] _skipped_packages+0x1e
	19.43%	python3	N/A	[.] 0x00007fbb0b3de5a2
	18.92%	python3	/usr/lib64/python3.6/site-packages/hawkey/__init__.py	[.] union+0xe
	2.07%	python3	N/A	[.] 0x00005570306a8674
	0.80%	python3	<frozen importlib._bootstrap_external>	[.] _compile_bytecode+0x0
	0.80%	python3	/usr/lib64/python3.6/site-packages/hawkey/__init__.py	[.] run+0x0
	0.28%	python3	/usr/lib64/python3.6/collections/__init__.py	[.] namedtuple+0x194
	0.26%	python3	<frozen importlib._bootstrap_external>	[.] find_spec+0x12
	0.26%	python3	/usr/lib64/python3.6/sre_parse.py	[.] _parse+0xfc
	0.23%	python3	N/A	[.] 0x00007fbafe413c12



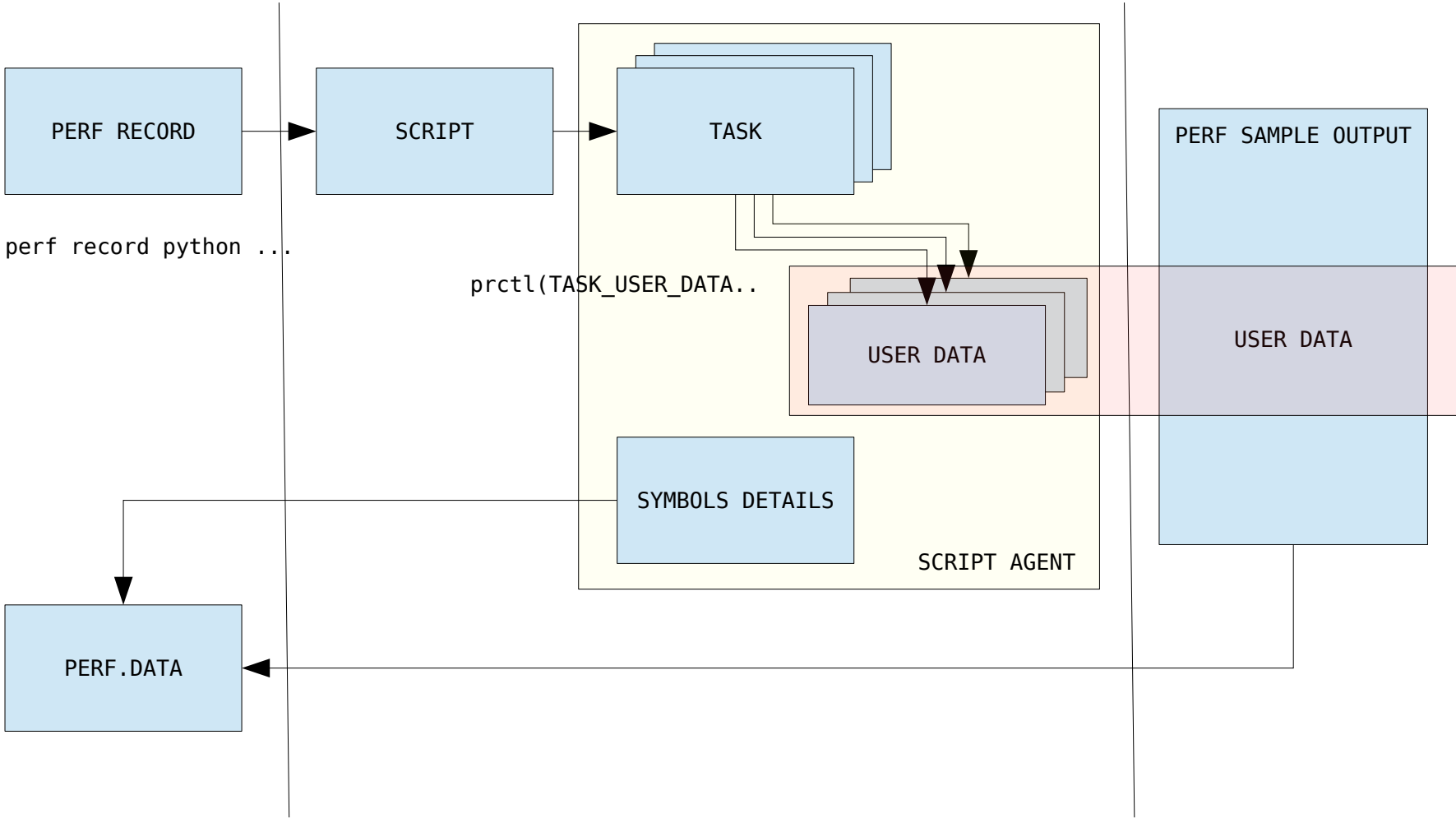
PROFILING SCRIPTS

Percent | Source code & Disassembly of [python_stack] for cycles:ppp (15 samples)
[getwidth /usr/lib64/python3.6/sre_parse.py]

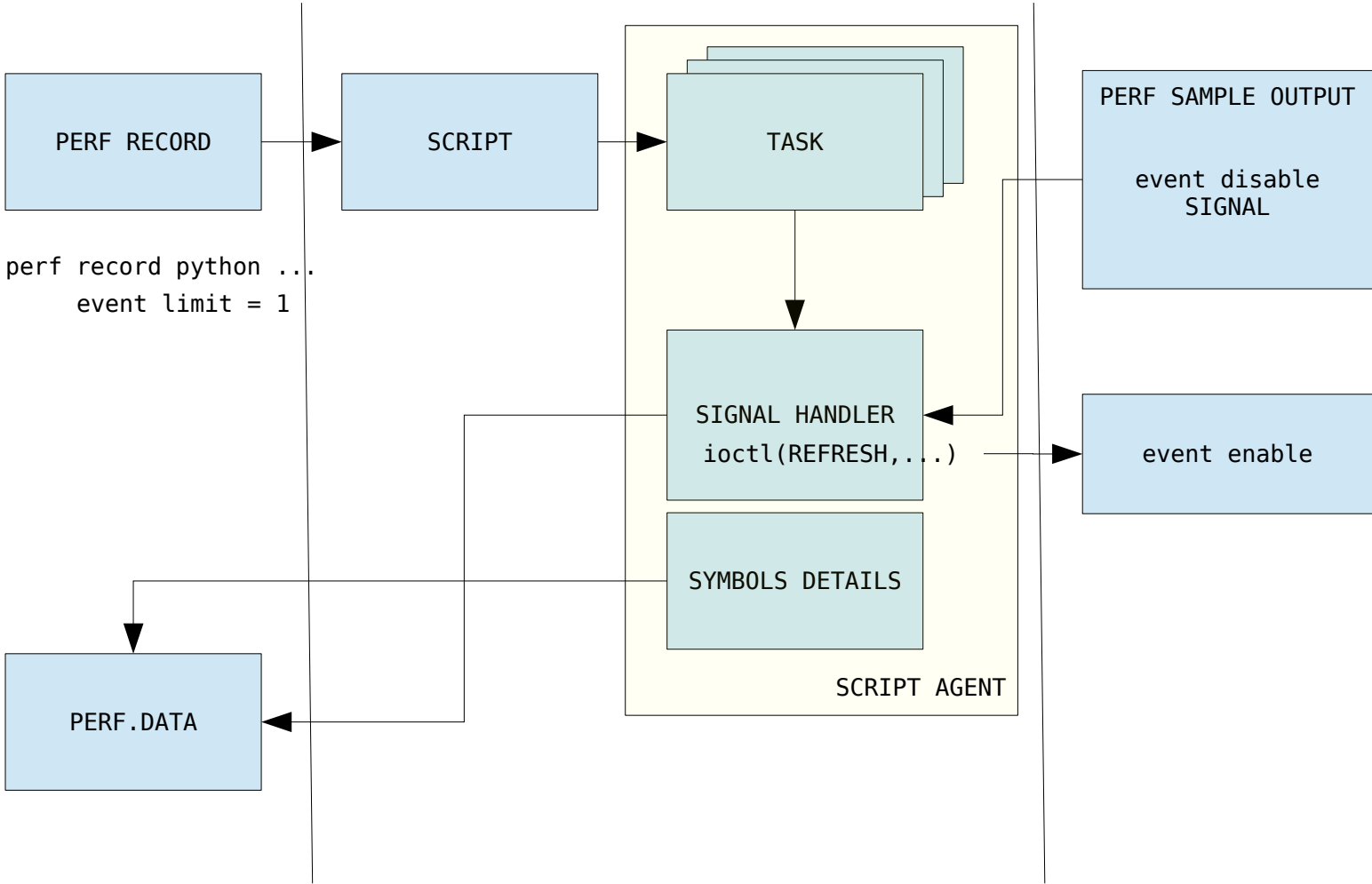
```
      :                               # determine the width (min, max) for this subpattern
0.00 : 55703057c4c0: if self.width is not None:
0.00 : 55703057c4ca:     return self.width
0.00 : 55703057c4d0: lo = hi = 0
6.67 : 55703057c4d8: for op, av in self.data:
0.00 : 55703057c4ec:     if op is BRANCH:
0.00 : 55703057c4f4:         i = MAXREPEAT - 1
0.00 : 55703057c4fc:         j = 0
0.00 : 55703057c500:         for av in av[1]:
6.67 : 55703057c50e:             l, h = av.getwidth()
0.00 : 55703057c51a:             i = min(i, l)
0.00 : 55703057c524:             j = max(j, h)
0.00 : 55703057c532:         lo = lo + i
0.00 : 55703057c53a:         hi = hi + j
6.67 : 55703057c544:     elif op is CALL:
0.00 : 55703057c54c:         i, j = av.getwidth()
0.00 : 55703057c558:         lo = lo + i
```



PROFILING SCRIPTS



PROFILING SCRIPTS



PROFILING SCRIPTS

- **currently works for python**
- **working on ruby & perl support**



PROFILING SCRIPTS

- source:

```
git://git.kernel.org/pub/scm/linux/kernel/git/jolsa/perf.git  
perf/script
```



THANKS, QUESTIONS

Jiri Olsa <jolsa@redhat.com>

