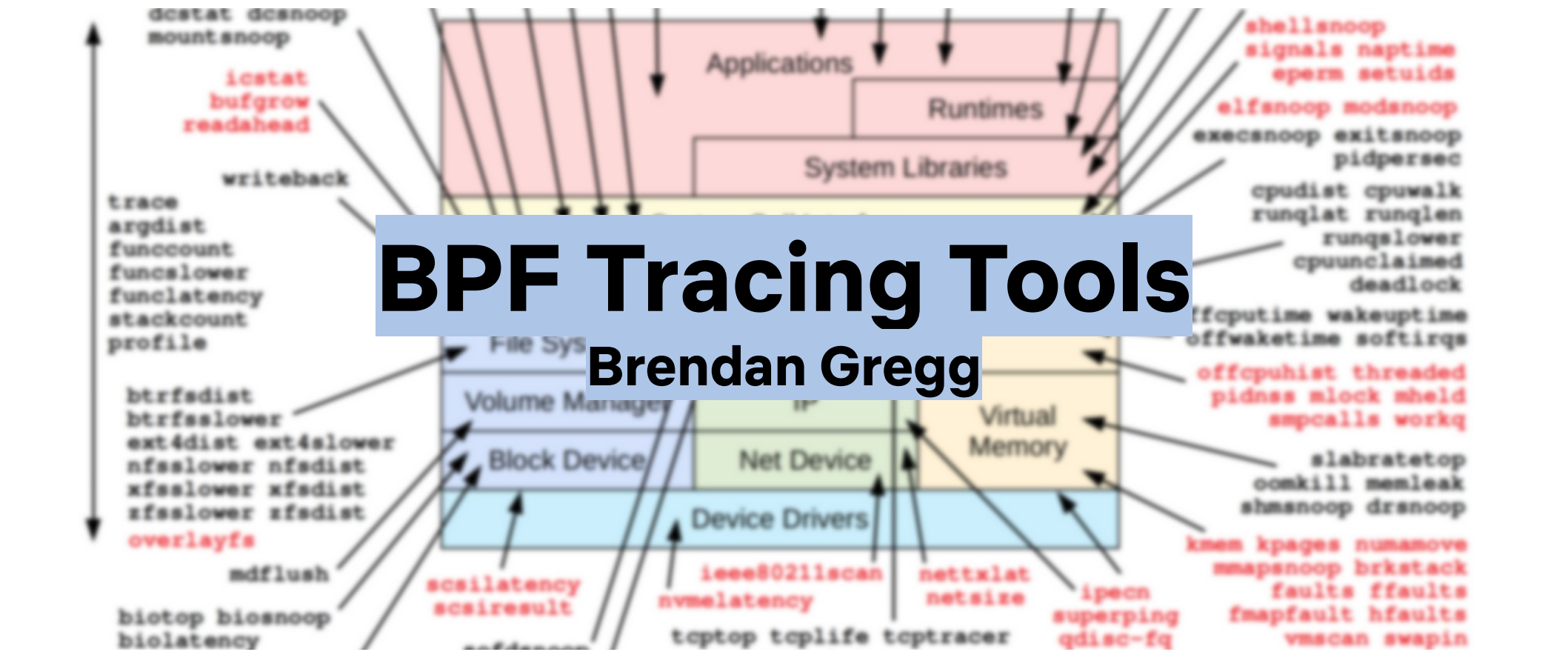
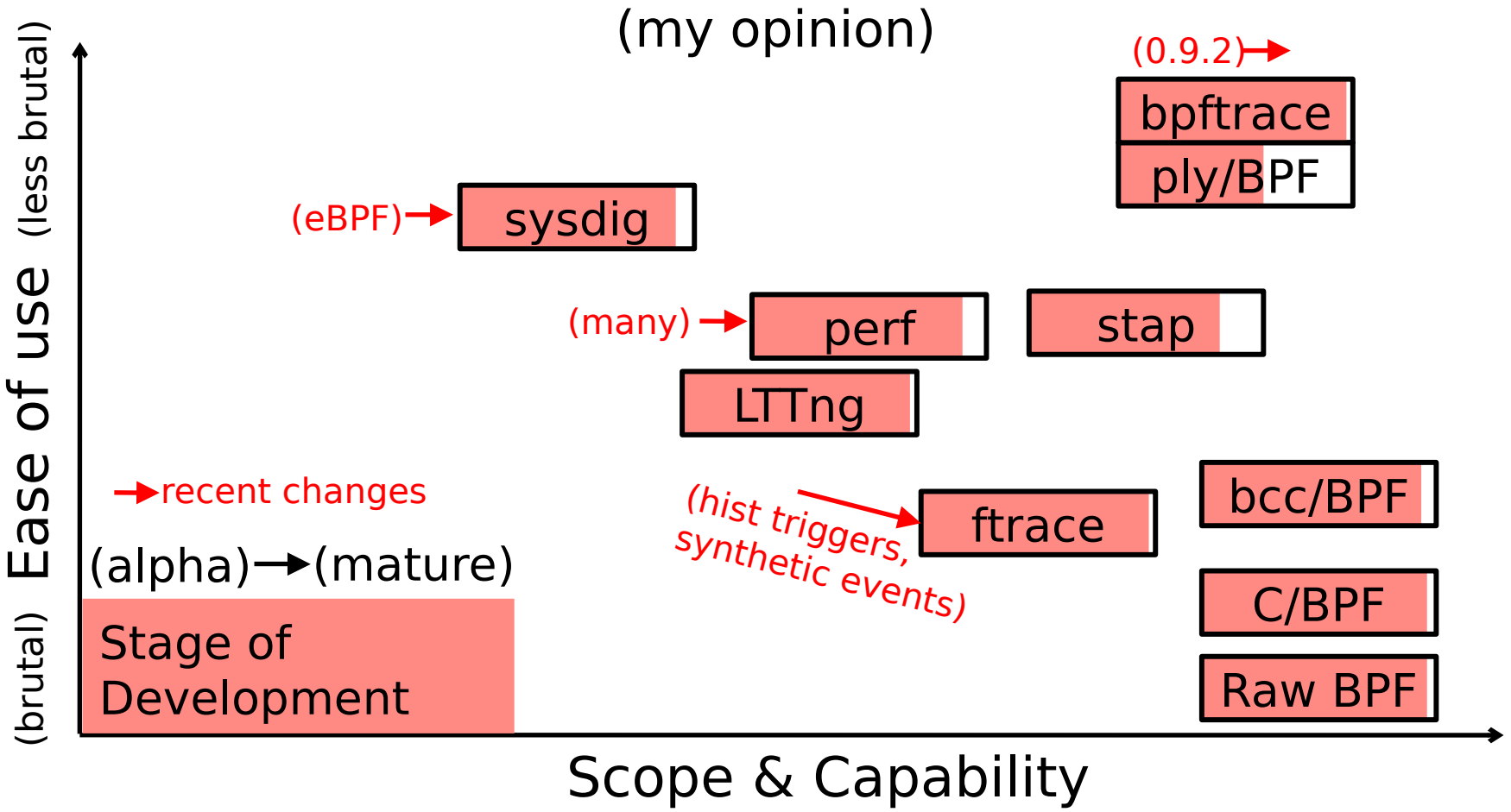


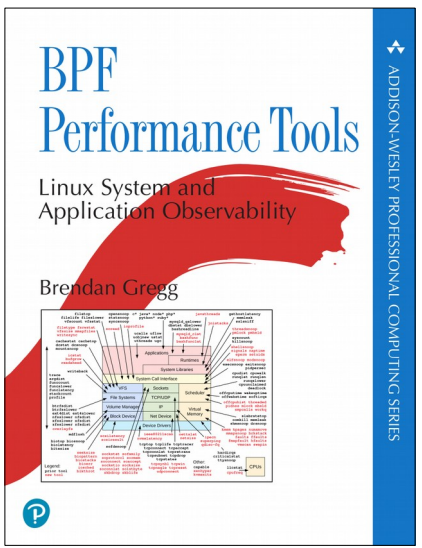
BPF Tracing Tools

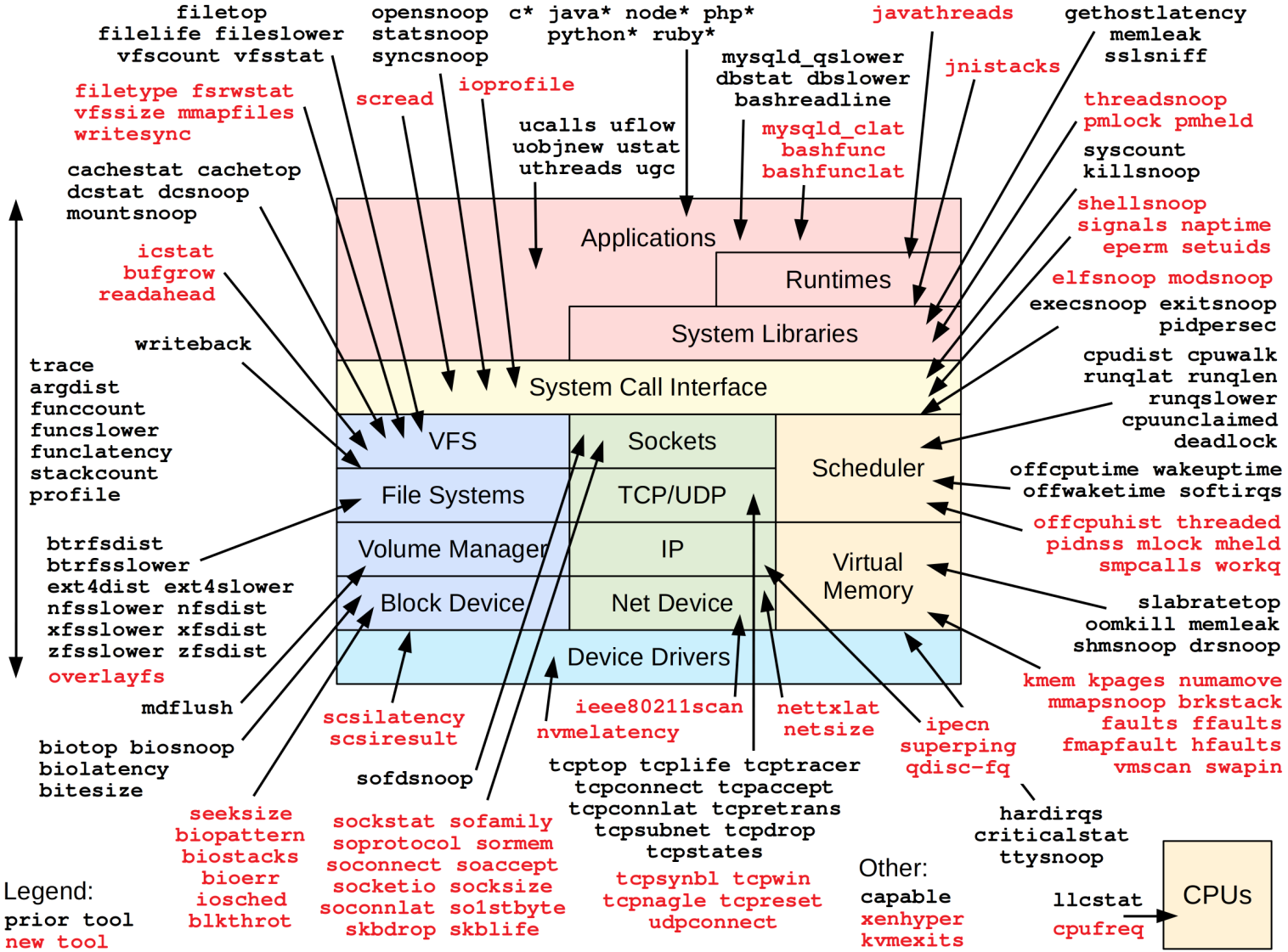
Brendan Gregg

The Tracing Landscape, Sep 2019



BPF Perf Tools





BPF Perf Tools Example: readahead

Is readahead polluting the cache?

```
# readahead.bt
Attaching 5 probes...
^C
Readahead unused pages: 128

Readahead used page age (ms):
@age_ms:
[1]                 2455 |@@@@@@@@@@@@@@@                                     |
[2, 4)              8424 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@|
[4, 8)              4417 |@@@@@@@@@@@@@@@@@@@@@@@@@@@                         |
[8, 16)             7680 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@     |
[16, 32)            4352 |@@@@@@@@@@@@@@@@@@@@@@@@@@@                         |
[32, 64)               0 |                                                    |
[64, 128)              0 |                                                    |
[128, 256)           384 |@@                                                  |
```

```
#!/usr/local/bin/bpftrace

kprobe:__do_page_cache_readahead    { @in_readahead[tid] = 1; }
kretprobe:__do_page_cache_readahead { @in_readahead[tid] = 0; }

kretprobe:__page_cache_alloc
/@in_readahead[tid]/
{
    @birth[retval] = nsecs;
    @rapages++;
}

kprobe:mark_page_accessed
/@birth[arg0]/
{
    @age_ms = hist((nsecs - @birth[arg0]) / 1000000);
    delete(@birth[arg0]);
    @rapages--;
}

END
{
    printf("\nReadahead unused pages: %d\n", @rapages);
    printf("\nReadahead used page age (ms):\n");
    print(@age_ms); clear(@age_ms);
    clear(@birth); clear(@in_readahead); clear(@rapages);
}
```

Discussion: Desired Tracepoints

- VFS
- socket send/recv, skb alloc to pair with skb:consume_skb/kfree_sub
- tcp send/recv, udp send/recv
- IP ECN
- genl, bql
- block:block_rq_issue/… add request pointer for use as unique ID
- locks

Discussion: Desired BPF Helpers

- struct file to pathname (like path_lookupat())
- FD to struct file / pathname / file type (DF_SOCK etc)
- bpf_get_current_pcomm()
- clock_gettime(CLOCK_PROCESS_CPUTIME_ID, ...)
- other timestamps
- more string functions

Discussion: Bigger Capabilities

- BTF (already there, thanks Yonghong Song etc)
- unprivileged BPF
- probe multi-attach (Ftrace is faster (__fentry__))
- faster uprobes (LTTng-style)
- bpf_probe_read_user/kernel split

Discussion: Challenges

- libc no frame pointer
  - LBR+FP stack walking (but no LBR on the cloud (mostly))
    update: user-level ORC a solution
- JIT function tracing