

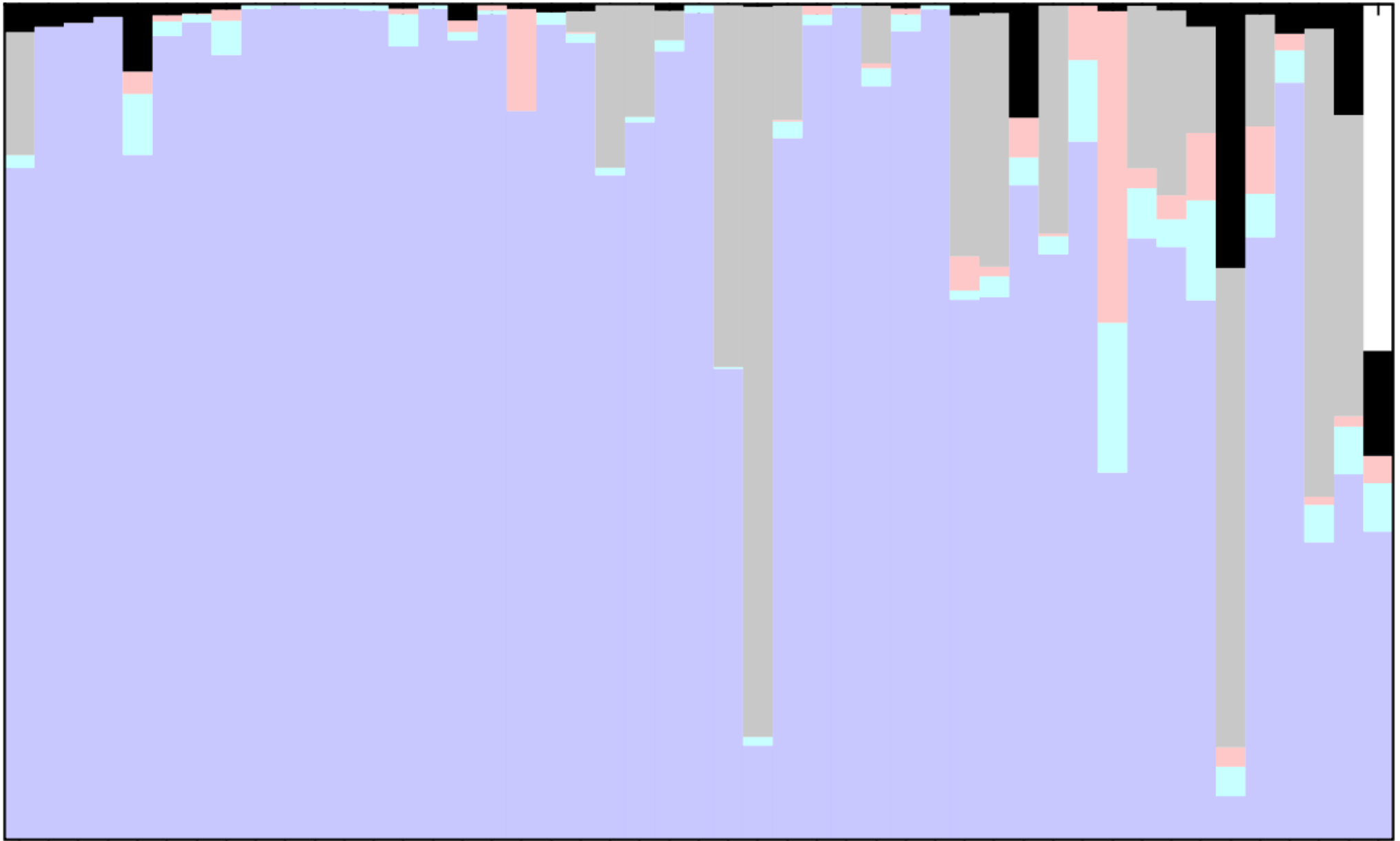
# Doing It Once

Automatically Improving Desktop Programs

Chris Snowton

# Desktop Programs Are Slow

- Start Firefox 3: 10s
- Start OpenOffice.org: 20s
- Mostly spent waiting on the disk



0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 3.2 3.4 3.6 3.8 4 4.2 4.4 4.6 4.8 5 5.2 5.4 5.6 5.8 6 6.2 6.4 6.6 6.8 7 7.2 7.4 7.6 7.8 8 8.2 8.4 8.6 8.8 9 9.2 9.4

B-self RK-self RU-self U-self others-others

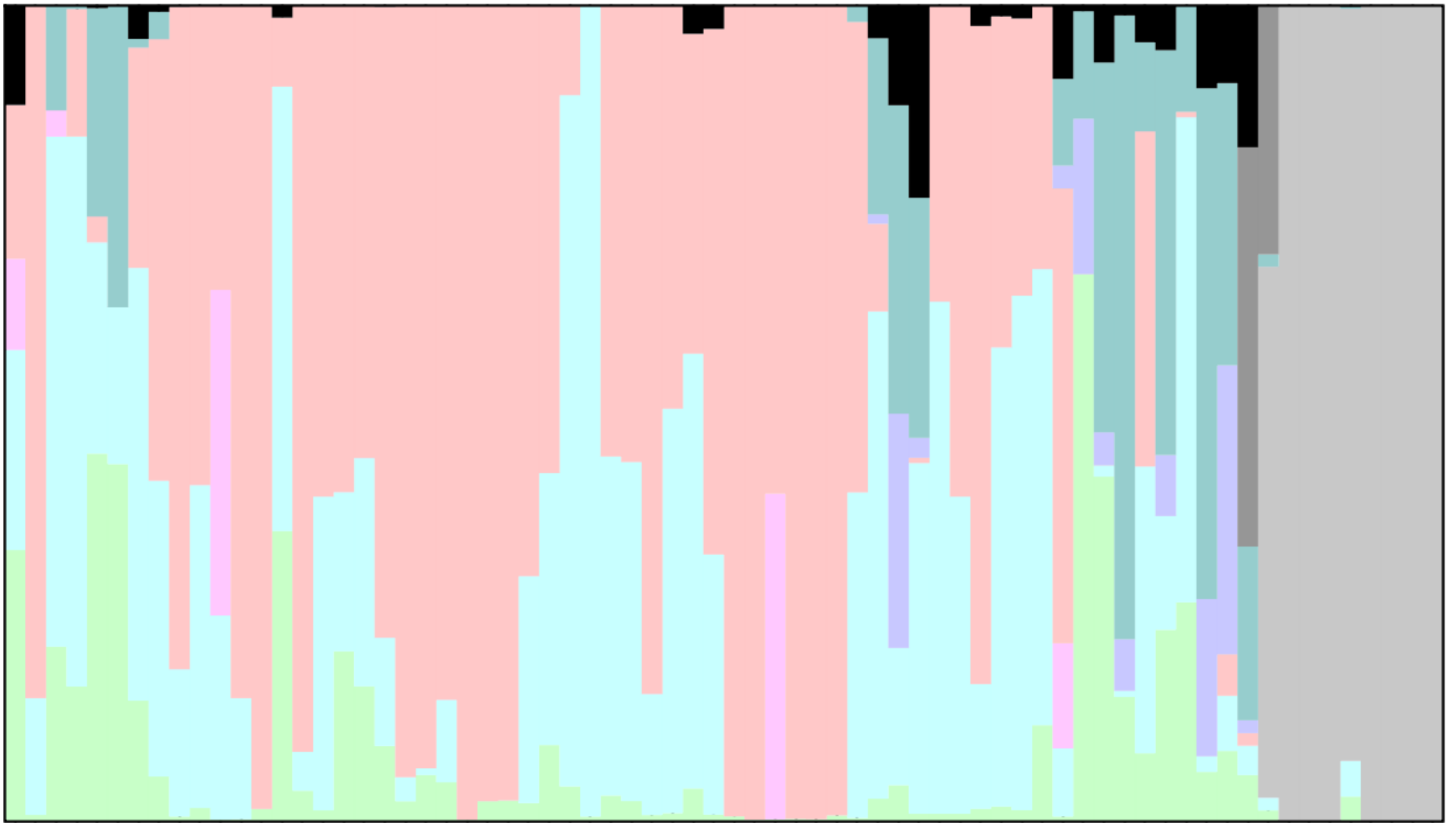


# But Haven't We Fixed That?

- Prefetch from disk
  - Using machine learning
  - Using compiler-generated hints
  - Using manual aio-fu

# Well, Kind Of

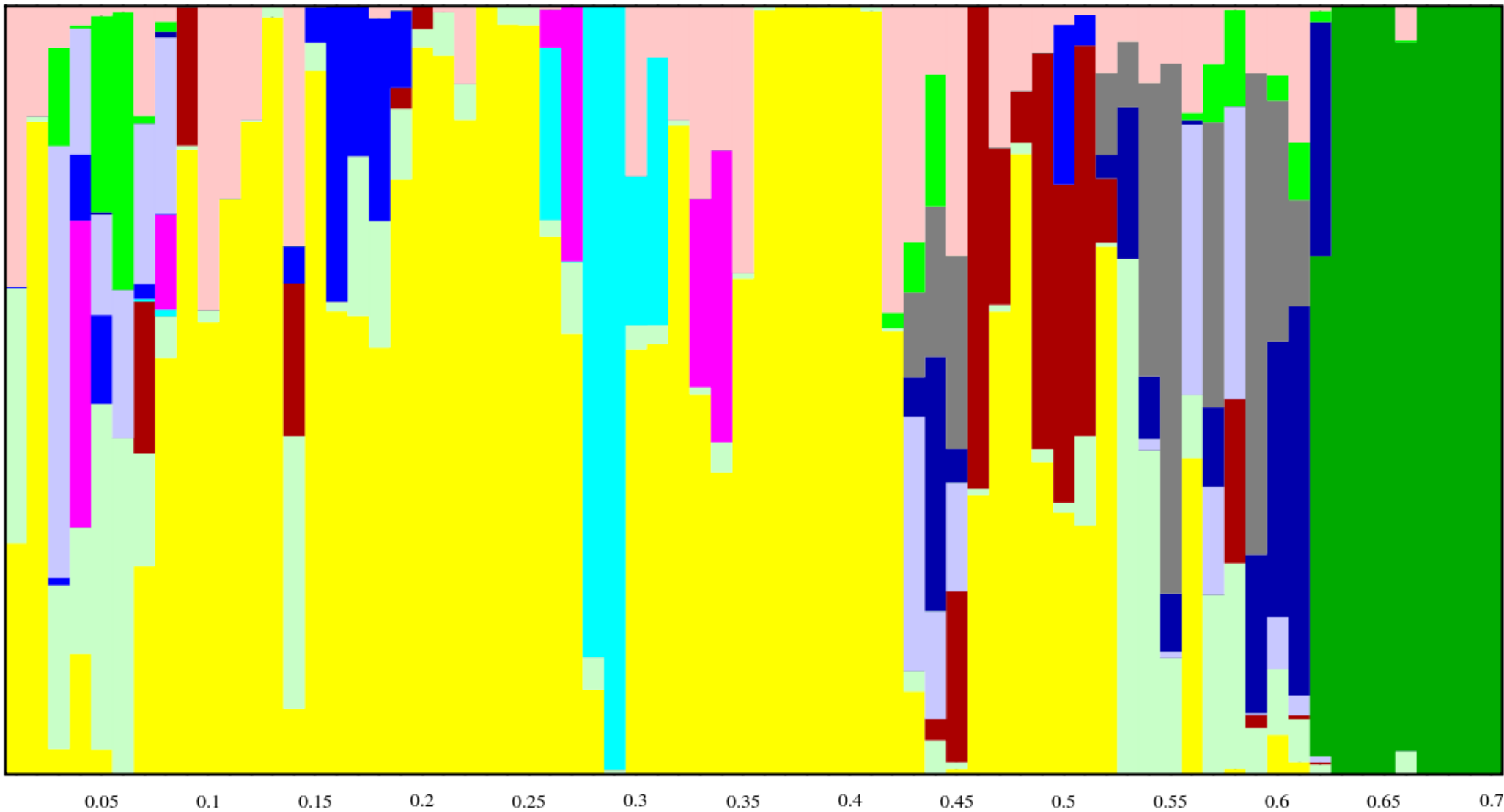
- Run with a warm cache: ideal prefetcher
  - OpenOffice.org: 3s
  - Firefox 3: 1s



0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55 0.6 0.65 0.7

- |         |  |   |  |               |  |
|---------|--|---|--|---------------|--|
| P-self  | <span style="color: #90EE90;">█</span> | RU-self                                 | <span style="color: #FFB6C1;">█</span> | RK-Xorg:3277  | <span style="color: #66CDAA;">█</span> |
| RK-self | <span style="color: #7FFFD4;">█</span> | U-/usr/lib/firefox-3.0.19/firefox:31474 | <span style="color: #A9A9A9;">█</span> | U-Xorg:3277   | <span style="color: #808080;">█</span> |
| RT-self | <span style="color: #FF69B4;">█</span> | B-Xorg:3277                             | <span style="color: #9370DB;">█</span> | others-others | <span style="color: #000000;">█</span> |





- |                            |   |   |  |
|----------------------------|---|---|--|
| other time (running)-self  | <span style="color: yellow;">■</span>     | sys_select+0x0/0x110-self                                 | <span style="color: lightblue;">■</span> |
| other time (waiting)-self  | <span style="color: lightgreen;">■</span> | other time (waiting)-usr/lib/firefox-3.0.19/firefox:31474 | <span style="color: green;">■</span>     |
| sys_futex+0x0/0x160-self   | <span style="color: darkred;">■</span>    | other time (waiting)-Xorg:3277                            | <span style="color: darkblue;">■</span>  |
| sys_getdents+0x0/0xf0-self | <span style="color: cyan;">■</span>       | sys_ioctl+0x0/0xe0-Xorg:3277                              | <span style="color: gray;">■</span>      |
| sys_open+0x0/0x30-self     | <span style="color: magenta;">■</span>    | sys_select+0x0/0x110-Xorg:3277                            | <span style="color: limegreen;">■</span> |
| sys_read+0x0/0xe0-self     | <span style="color: blue;">■</span>       | others-others   | <span style="color: pink;">■</span>      |

# How can we make it faster?

- Assertion: much work is repeated from run to run
  - Example: parse the same file again
  - Example: transform format of same data
  - Example: link the same libraries again

# But haven't we fixed **that**?

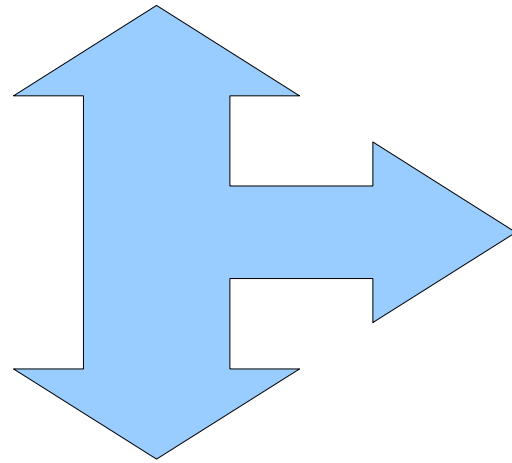
- Some of it: `prelink`
  - Pre-prepare linked images
  - Move work away from runtime
    - In particular **repeated** work

# How can we do more?

- Partial evaluation
  - Specialising programs
    - According to the contents of some file
    - According to the return from `uname`
    - According to our current network setup
- Informed speculation
- Key theme: **move work away from runtime**

# Partial Evaluation

$f(x, y)$



$f_{42}(x)$

$y = 42$

# Practical Partial Evaluation

```
c = read_all("/etc/myapp.conf");  
s = conf_to_state(c);  
main_loop(s);
```

# Warm the cache...

```
c = read_all("/etc/myapp.conf");  
s = conf_to_state(c);  
main_loop(s);
```

# Apply partial eval...

```
c = read_all("/etc/myapp.conf");  
s = conf_to_state(c);  
main_loop(s);
```

# Not just for startup...

```
main_loop(State s) {  
    while(true) {  
        i = get_input();  
        if(s.ssl_enabled) {  
            ...  
        }  
        else {  
            switch(s.minor_mode) {  
                ...  
            }  
        } ...  
    } ...  
}
```

# Not just for startup...

```
main_loop(State s) {  
    while(true) {  
        i = get_input();  
        if(s.ssl_enabled) {  
            ...  
        }  
        else {  
            switch(s.minor_mode) {  
                ...  
            }  
        } ...  
    } ...  
}
```

# Informed Speculation

- Why speculate?
  - Waiting for disk
  - Waiting for Xorg
  - Waiting for network
- Informed?
  - Augment binary with **hints**

# Hints?

```
handle_event() {  
    x = send_request(...);  
    r = await_reply(x);  
    if(r.is_success) {  
        /* expensive work... */  
    }  
    else {  
        ...  
    }  
}
```

# Hints!

```
handle_event() {  
    x = send_request(...);  
    r = await_reply(x);  
    if(r.is_success /* HINT true */) {  
        /* expensive work... */  
    }  
    else {  
        ...  
    }  
}
```

# What else might we hint?

- Important subsets of files
  - e.g. headers of ELF file are important
  - Package them with the binary as **hints**
    - Speculate whilst we wait for disk

# Summary

- Make desktop software faster
- By pushing work away from runtime

Questions?