Doing It Once
Automatically Improving Desktop Programs

Chris Smowton
Desktop Programs Are Slow

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

\[ f(x, y) \]

\[ y = 42 \]

\[ f_{42}(x) \]
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);
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?