Doing It Once
Automatically Eliminating Redundant Work

Chris Smowton
Programs Are Bad

• Recompute the same old stuff
  • In the same run
  • Between runs
Programmers Are Bad

- Optimise until people stop complaining
- But... unanticipated situations?
  - Slow machines
  - Heavily loaded machines
  - Used embedded within another program
Compilers Are Good

- Automatic program improvement
- At no cost

\[ x = \arctan(y); \]
\[ z = \sinh(w) + \arctan(y); \]
\[ t = \arctan(y); \]
\[ x = t; \]
\[ z = \sinh(w) + t; \]
Compilers Are Limited

- Need the source
- Compile-time constants only
- Seldom use feedback
  - Better: LLVM
So: Let's Improve Binaries

- Remove redundant work:
  - Reading same file
    - As earlier in this run
    - As on previous runs
  - Predictable conversation with Xorg
  - Predictable thread interleaving (--> coroutines)
Haven't We Already Done That?

- Page cache
  - Eliminates much of FF's startup:
What's to Gain?

● Long disk waits
  • Solved: page cache

● First time still slow (cold cache)
  • Solved: ReadyBoost, ureadahead

● Time spent parsing / building in-memory structs?
  • Snapshots?
  • Partial evaluation!
Partial Evaluation

\[ f(x, y) \]

\[ y = "dave" \]

\[ f\_dave(x) \]
Partial Evaluation

- Treat file contents as constants
- Propagate constants
- Inline functions
- Unroll loops
- --> Faster startup
- --> Faster at runtime
- --> Smaller binary
- make firefox_config
And More

- Specialise for common situations
- Specialise “cooperative” processes
  - Xorg, gconf
- Informed speculation
Questions?