Storing Path Information in Crash Reports

Joel Winstead

When a program crashes,
we want as much information as possible to isolate the failure.

It is not practical to record a complete trace for a deployed application:
disk space, bandwidth, performance, privacy constraints.
We typically use a compact crash report instead, including:
- Point of failure
- Register contents
- Stack trace
- Memory dump

It is not always obvious from the crash dump
how the program reached the point of failure:
we must reason backwards from the final state.

We could include information about recent branches taken:
This could give us the path taken through the procedure
in which the failure occurred.
This information may help isolate or reproduce the failure.
Combined with static analysis, this may help reconstruct
the program's behavior leading up to the failure.

Low overhead:
Branch predictors already track this in hardware.
We could also do this by instrumenting the program.
One or two bits per branch would be compact.

Challenges:
One or two bits per branch would underdetermine behavior.
We can only afford to store recent behavior, not entire execution.
We may need to know more about program's state to isolate problem.