

# Handling Self-Modifying Code Using Software Dynamic Translation

Joy W. Kamunyori  
jwk7z@cs.virginia.edu

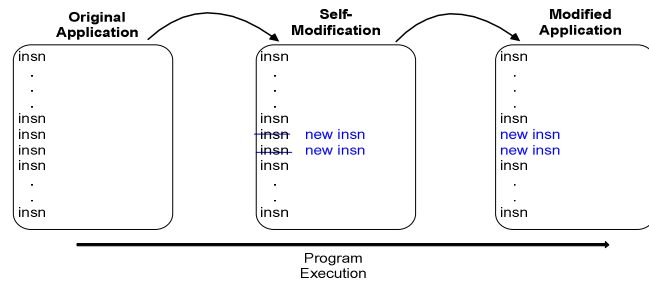
## The Challenge

Because Strata caches instructions, it must be made aware that a previously translated instruction has changed and should be re-translated.

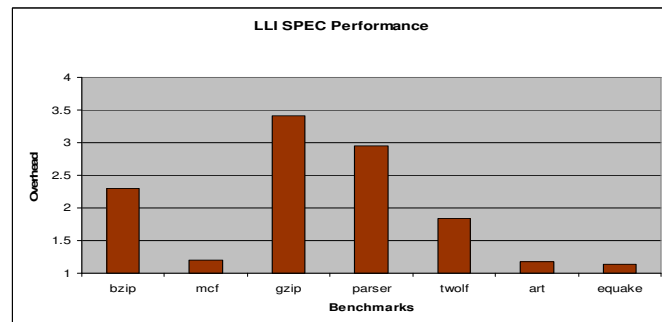
## What is Self-Modifying Code?

**Self-modifying code** refers to code that changes itself during a program's execution. Applications such as Just-In-Time (JIT) compilers overwrite their own code as a matter of routine.

Self-modifying code can be used for malicious purposes, so it is important to be able to handle it in a safe, prescribed manner.



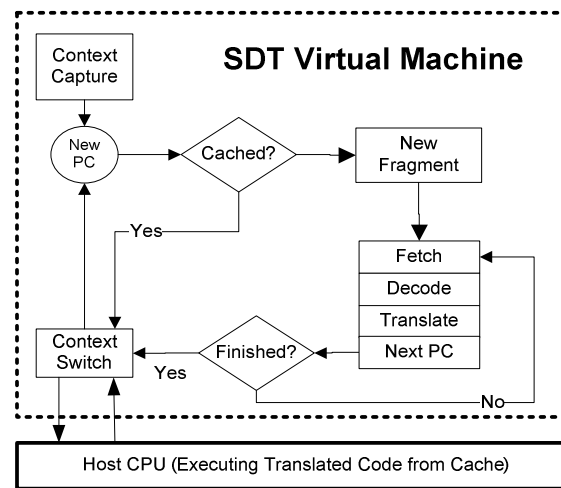
## Results<sub>1</sub>



1. Results derived using LLI, the JIT compiler in the Low Level Virtual Machine (LLVM) developed at the University of Illinois at Urbana-Champaign

## What is Strata?

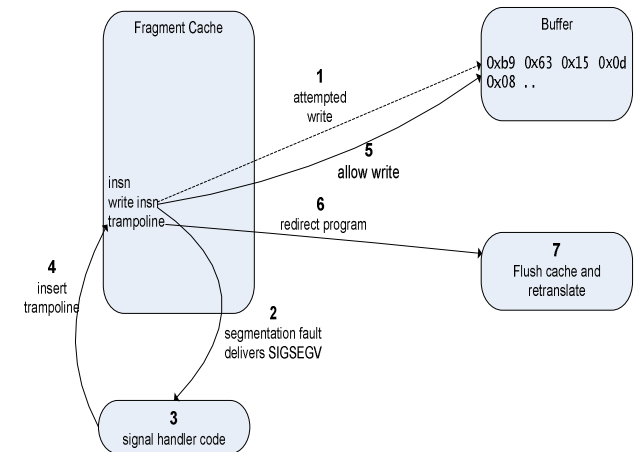
**Software Dynamic Translation** rewrites a program's execution binary at runtime. **Strata** is a portable, extensible SDT framework, that runs on x86/Linux and Sparc/Solaris. Strata mediates program execution transparently, and with low overhead.



## How Strata Works

- Application instructions are translated and stored in the fragment cache. Each instruction is examined before translation to ensure that it has not been previously translated. Previously translated instructions are not re-translated.
- Once a fragment has been fully created, the newly-translated fragment is executed.
- If a previously translated application instruction is encountered, Strata executes the instructions stored in the fragment cache.

## High-Level Algorithm



Enforcing instruction retranslation involves the following steps:

- **Turn off write permissions:** The application text is write protected, and all calls to mprotect are intercepted to ensure that write permissions are not turned back on.
- **Intercept signals:** The segmentation fault caused by an attempt to write (1) to the protected text is intercepted (2) and sent to a signal handler (3).
- **Allow write:** The write permissions are turned back on so that the attempted write can occur (5).
- **Flush cache and retranslate:** The trampoline inserted by the signal handler (4) redirects the program (6) to code where the fragment cache is flushed and all instructions following the changed instruction are retranslated (7).