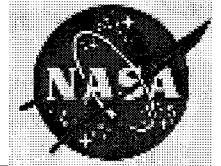


Assessing Application Vulnerability to Radiation-Induced SEUs in RAM

Paul L. Springer



Fault Model



JPL's Remote Exploration and Experimentation Project (REE) wants to:

- **Use COTS computers in space**
- **Understand SEU effects on applications**

First understand radiation-induced SEUs in RAM:

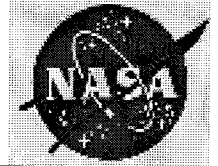
- **Random distribution in memory space**
- **Random distribution in time**

Only some SEUs in RAM impact application

- **For three REE applications, code and data fault injections:
90% of faults had no effect
The main reason: RAM location not vulnerable during injection**

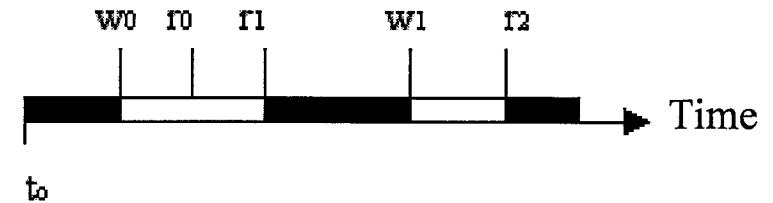


Measure of Vulnerability



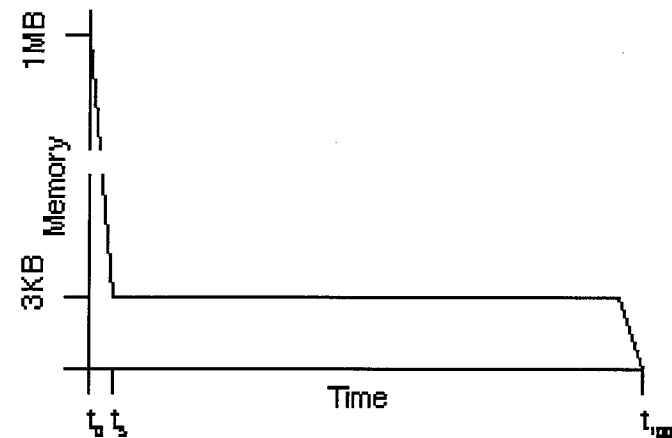
Data Vulnerability (V_d)

- Variable is written to at times w_0 and w_1
- Variable is read at times r_0 , r_1 , and r_2
- Variable is vulnerable between w_0 & r_1 , and w_1 & r_2
- V_d is sum of all vulnerable times over all bytes



Code Vulnerability (V_c)

- Code byte is vulnerable from time it is loaded until the last time it is executed
- Code vulnerability profile plots number of vulnerable code bytes against runtime
- Profile is typically non-increasing
- V_c is integral of profile

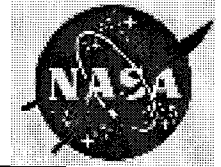


Vulnerability Metric

- Sum of vulnerability times over all bytes, in units of MB-seconds
- Code (V_c) and data (V_d) vulnerabilities can be combined or used separately



Usefulness of Metric



Not impossible to obtain

- Finding approximations or bounds on V_c and V_d much easier than measuring every byte's vulnerability, and can be as useful

Important determinant in predicting application susceptibility to SEUs

Use to compare vulnerabilities of two different program implementations

Analyze tradeoffs of adding ABFT code

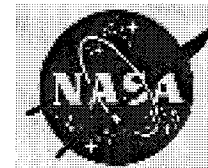
- ABFT effectively decreases V_d , but increases V_c

Characterize how parallelism affects vulnerability

- V_c : For this MB-seconds measure, MB increases (multiple nodes), but runtime decreases; the profile also changes
- V_d : MB about the same (distributed data), runtime decreases
- Parallelism may increase or decrease total vulnerability



Future Work



Develop tools to help measure vulnerability profiles

- **Similar in nature to some performance tools**

Construct guidelines for decreasing vulnerability

Characterize parallelism effects on vulnerability

Study ABFT effects on total vulnerability

- **Are there times when ABFT might be counter-productive?**

Determine if any of this work can be applied to analyzing SEUs in processor cache

<http://hpc.jpl.nasa.gov/PEP/pls/slides/DSN2001.html>