The goal of our work is to develop a trustworthy, flexible and efficient platform for reliable computing in the presence of transient faults.

- Transient faults (also known as soft fails) occur when an energetic particle strikes a transistor, causing it to change state.
  - Does not permanently damage hardware.
  - May corrupt computation by altering stored values and signal transfers.

\[ 1 + 1 = 18 \]

- Sun, HP, and Cypress Semiconductor have admitted that transient faults caused crashes and problems at eBay, AOL, Los Alamos, and other major sites.
- Faster clock rates, increasing transistor density, decreasing voltages, and smaller feature size all contribute to increasing fault rates of approximately 8% per generation.

- There are many existing solutions, but do these solutions actually work?
  Solutions often consist of algorithms stated in English, and it is left to the audience to judge correctness.
- How can existing solutions protect applications in an efficient manner?
  Solutions often provide uniform protection regardless of native reliability, performance cost, and protection benefit.

**Approach and Impact**

<table>
<thead>
<tr>
<th>New approach</th>
<th>Research Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Formal typing rules for redundant computation</td>
<td>• Provide <strong>provable</strong> fault protection</td>
</tr>
<tr>
<td>• Software-modulated protection to precisely tradeoff between performance and reliability</td>
<td>• Provide <strong>efficient</strong> fault protection</td>
</tr>
</tbody>
</table>

- **TAL\text{\textregistered}** : Detect all faults that change a program’s observable behavior and prove that *well-typed programs always detect a single fault*.
- Operational semantics \( \Sigma \rightarrow \Sigma' \) specifies how a program executes.
  - Special operational rules model faults in the processor data paths
  - Judgment \( \vdash^Z \Sigma \) states that machine state \( \Sigma \) is well-typed under zap tag \( Z \).
  - *(Simplified) Fault-Tolerance Theorem:* Either a faulty execution is indistinguishable from the equivalent fault-free execution, or it terminates in the *fault* state.

  If \( \vdash \Sigma \) and \( \Sigma \sim^Z \Sigma_1 \) and \( \Sigma \rightarrow \Sigma' \)
  then either \( \Sigma_1 \rightarrow \Sigma_2 \) and \( \Sigma' \sim^Z \Sigma_2 \)
  or \( \Sigma_1 \rightarrow \text{fault} \).

- **SMFT** : For a given budget provide most reliability and best performance by only enabling protection for application regions that are critical for correct execution.

  • By modulating the protection level, can trade off reliability for performance

**Representative Publication:**


Winner of the PLDI 2007 Best Paper Award.