Tic Tac Toe Game Performance Report

Executive Summary

This report provides a comprehensive analysis of the Tic Tac Toe game's performance across various operational states, including general gameplay and the login sequence, based on collected CPU, Disk I/O, and Memory usage data. The overall findings indicate that the game consistently demonstrates lightweight resource consumption, posing minimal impact on system performance during the observed periods.

Detailed Analysis

1. CPU Usage

- **Observation:** The data indicates that the Tic Tac Toe game consistently utilizes **0% of the CPU** during login page state, with an average CPU usage of **0.10%**. It operates with **11 threads**. During the general operation, it also registers **0% CPU utilization**, with an average CPU usage of **0.09%**, and operates with **7 threads**.
- **Analysis:** Both general gameplay and the login sequence show exceptionally low CPU consumption. This suggests that the application is highly optimized for minimal processing, remains largely idle during these states, or performs background tasks very efficiently. Its impact on the CPU is negligible across observed scenarios.

2. Disk I/O

- **Observation:**
    - One observation shows the game with **0 B Read (B/sec)** and **1,130 B Write (B/sec)**, totaling **1,130 B (B/sec)**.
    - Another observation indicates the game with **0 B Read (B/sec)** and **596 B Write (B/sec)**, totaling **596 B (B/sec**
- **Analysis:** The disk activity for the Tic Tac Toe game is consistently minimal, primarily involving small write operations directed to a system log file. There are no significant read operations observed across the provided data. This low disk I/O suggests that the application is not heavily reliant on disk access for its operations, which is beneficial for overall system responsiveness. The consistent 0 response time further indicates that these minor operations are handled instantaneously by the system.

3. Memory Usage

- **General Gameplay Observations:**
    - **Observation 1:** Commit (KB): 14,608; Working Set (KB): 43,904; Shareable (KB): 32,956; Private (KB): 10,948.
    - **Observation 2:** Commit (KB): 14,536; Working Set (KB): 44,312; Shareable (KB): 33,796; Private (KB): 10,516.
    - **Observation 3:** Commit (KB): 14,640; Working Set (KB): 45,640; Shareable (KB): 34,572; Private (KB): 11,076.
    - **Observation 4 (during reply):** Commit (KB): 15,140; Working Set (KB): 46,196; Shareable (KB): 34,632; Private (KB): 11,564 (with 2 Hard Faults/sec).
- **Login Page Observation:**
    - **Observation:** Commit (KB): 14,926; Working Set (KB): 45,152; Shareable (KB): 33,668; Private (KB): 11,484 (with 1 Hard Fault/sec).
- **Analysis:**
    - **Commit (KB):** The committed virtual memory for the Tic Tac Toe game consistently ranges from approximately 14.5 MB to 15.1 MB across all observed scenarios, including the login page. This indicates a relatively small and efficient reservation of virtual memory by the operating system for the application.
    - **Working Set (KB):** The physical memory (RAM) actively used by the process fluctuates moderately between roughly 43.9 MB and 46.2 MB. This usage falls well within acceptable limits for most applications and suggests efficient RAM management.
    - **Shareable (KB) vs. Private (KB):** A significant portion of the working set (around 33-34 MB) is shareable memory, allowing it to be reused by other processes, which enhances overall system memory efficiency. The private working set, exclusively used by the application, remains small (approximately 10.5 MB to 11.5 MB), which is a positive indicator of lean memory allocation.
    - **Hard Faults/sec:** The application shows a very low rate of hard faults (1 to 2 faults/sec) in some observations. While hard faults indicate that the system

had to retrieve a memory page from disk, these numbers are negligible and do not suggest a performance bottleneck or memory pressure issues.

Conclusion

Based on the comprehensive performance metrics, the Tic Tac Toe game consistently demonstrates excellent resource efficiency across general usage and specific states like the login page:

- **CPU Usage:** Consistently extremely low, indicating minimal processing overhead and high efficiency.
- **Disk I/O:** Very low, primarily for background logging, suggesting efficient and non-intensive data handling.
- **Memory Usage:** Moderate and exceptionally well-managed, with a significant proportion of shareable memory and an insignificant rate of hard faults.

The game appears to be robustly optimized and does not impose any discernible resource bottlenecks on the system, providing a smooth user experience from a resource consumption perspective.