Maximizing CICS Performance with Observability Insights

An Intellyx White Paper for Rocket Software
by Jason 'JE' English, Director & Principal Analyst

January 2025

Introduction

Over the last twenty years, we've seen a quiet revolution in the performance engineering space. Early user benchmarking and server monitoring tools gave way to advanced application observability platforms that can absorb vast quantities of application telemetry data to feed developer and SRE investigation and performance tuning processes.

Amidst such advances, the mainframe, the beating heart of the organization and the backbone of the global economy, has remained a black box to most of the IT team at large as they attempt to move the application estate forward.

Even if leadership tries to reallocate resources to cover core systems, only the most experienced mainframers are ready to dig inside a CICS region and figure out why a system isn't working. Developers don't have the means to investigate why queries take too long, or transactions time out–they only know that something isn't working as well as the customer expected.

This paper will discuss how CICS observability offers unique challenges and opportunities to the performance-minded enterprise. Using a solution like Rocket Software's C\Prof enables mainframe performance teams to improve performance, reduce risk and save costs across the application estate, with quality and performance levers and visibility that even non-expert stakeholders can understand and learn from.

The Challenges of Compounding Resource Usage

Mainframes are critical to enterprises around the world, processing billions of transactions every day. Mainframe resource usage will only increase as more connected Internet services and customer front ends go online, further compounding the need for higher back-end scalability without costly downtime risks.

A typical mainframe outage can cost an enterprise millions of dollars an hour. A recent Forbes article noted that a cost of US$9,000 per minute of downtime is not unusual, and Forrester reported that for a real-time industry such as a brokerage service, one hour of downtime could cost as much as $6.48M an hour.

To face down such massive economic damages, regulatory bodies and industry consortiums are advancing compliance regimes and standards, such as DORA, for mitigating costly failures. While this is important for the future, it can also lead to a change avoidance mentality, where development and operations teams prioritize 'not breaking anything' over further innovation---possibly an even more costly long-term situation.

Finally, there's a massive talent crunch affecting mainframe teams, as the most experienced engineers move on in their careers. There are never enough SMEs ready and willing to chase down problems and work out enhancements that would improve scalability.

Application optimization atop CICS is hard, so we need common observability and reporting tools that will help newer mainframe talent contribute value faster, while bringing other application teams on board with the visibility and feedback needed for success.

Earlier Observability for Everyone

For as long as we've had enterprise systems, operations teams evaluated themselves by availability and transactional consistency. If downtime or issues arise, then MTTR (mean time to repair) and other MTTs such as MTTA (acceptance), MTTU (understanding), and so on, become critical metrics for evaluating the team's incident response times.

While such MTT* metrics are useful, they are essentially looking backward at production systems, which provides little guidance for improvement going forward.

What could we accomplish by shifting CICS observability left into pre-production?

- **Performance optimization** by testing and identifying issues even in pre-production, and providing production issue feedback to inform DevOps lifecycles, in order to quickly get down to root causes and improve long-term performance.
- **Reduced operating cost** and better utilization of CPU and storage resources thanks to early visibility into consumption trends and detection of anomalies.
- **Better division of labor** to align both mainframe SMEs and non-mainframe talent to get involved in managing performance for a wider variety of target applications.
- **Better customer experience** by setting SLOs (service level objectives) for better system response times and improved resiliency that DevOps and SRE teams can understand as part of their design and larger DevOps initiatives.

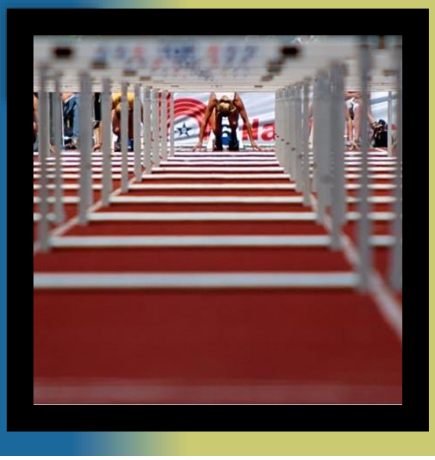
Setting Observability Objectives with C\Prof

Rocket Software's C\Prof solution provides early analysis and diagnostics to trace individual transactions and clearly identify problems within CICS applications, without requiring any code or agent insertions into CICS that could impact business-critical applications.

Observation of trends over time is the key to early recognition, and C\Prof provides CICS trace information in an easy-to-read format with clear data visualizations, while still providing access to the actual trace information of individual transactions happening across multiple CICS regions.

Systems programmers with more mainframe experience can do deep introspection on events with point-in-time snapshots of CICS internal trace data within C\Prof.

From there, expert mainframers can either resolve issues on the spot, or if there is a dependency, clearly reproduce issues for other application developers and operators so they can quickly and collaboratively kick off remediation tasks or tickets in their own service management tools.



Using these capabilities, mainframe engineers, development teams, and business stakeholders can come together and align their efforts around achieving observability SLOs, for example:

- **Cost savings:** Cost-per-transaction and system utilization ratios can be improved by identifying CICS tuning opportunities to reduce CPU overhead, reduce monthly costs, and increase capacity only when demand surges. Cost savings also come from decreased power usage, which contributes to sustainability goals.
- **Mean time to repair:** How long does it take someone with responsibility to identify, accept and resolve issues? Fast problem diagnosis and remediation

strengthens operational stability and reduces costs related to downtime or outages. Set expectations for continuous incremental improvement in this metric, bearing in mind that when more minor issues are being discovered and repaired, that may be a good thing, since they won't become costlier and time consuming Sev 2 or Sev 1 issues.

- **Quality of experience (QoE):** Customers and end users only notice back-end latency and interruptions when they become noticeable within downstream applications. Application teams should define a portion of their overall performance budget to include CICS response times and alert the team if it goes beyond that threshold. In addition, set tighter goals to squash any functional bugs or failures as quickly as possible.

- **Time on task / reduced toil:** Some nuance is required here, since we aren't measuring engineering 'productivity' so much as attempting to reduce the amount of time spent on unproductive toil such as manually searching for specific transaction errors, or attempting to report issues only to have them closed early by another team who couldn't reproduce the issue. The best technical talent wants to spend more time innovating and contributing value, so reducing toil will also improve employee satisfaction and retention.

The above SLOs are just general ideas. To realize value from observability initiatives, they should be tuned toward the operational realities of the specific enterprise and the industry, as the following examples demonstrate.

Example CICS Observability Outcomes

Retail Bank

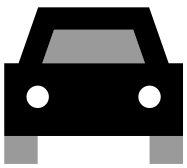
Problem: With senior members of this financial institution's once-stable mainframe team approaching retirement, newer team members were having difficulties stepping in and understanding CICS application issues.

Solution: Using C\Prof, they were able to aggregate CICS trace information from across multiple regions into a combined view, giving newer engineers metrics that could drill down into individual transactions, thereby reducing MTTR and business disruptions without needing to allocate additional headcount.

Healthcare Organization

Problem: This organization had built up a variety of homegrown solutions to handle specialized business workflows atop multiple CICS regions, but teams of mixed experience levels started having difficulties supporting high availability standards atop existing architectures as transaction volumes increased.

Solution: Using C\Prof to identify exact commands that were generating non-essential system work, the team eliminated 1 billion non-essential transactions per week across three file owning regions (FORs). The analysis process placed no resource burden on production systems, while allowing the firm to save up to 25% of z/16 processor time and improve transaction throughput speeds by 50 percent.



Regional Auto Club

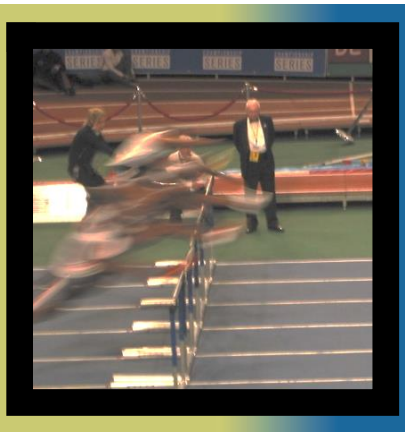
Problem: CICS system performance and response times for the company's heavily used transaction system were flagging, as CPU consumption levels mysteriously increased during peak hours, requiring expensive capacity-on-demand (CoD) expenditures to meet service level agreements (SLAs).

Solution: By using C\Prof to collect and forensically analyze CICS traces without running inside the region, they identified and eliminated abends within a particularly poorly performing transaction—resulting in more than a 90% decrease in CPU time and associated CoD charges for that application, for a projected monthly savings of as much as $300K.

The Intellyx Take

By the time an end customer complains about how an enterprise application is slow or buggy, they don't care which faulty CICS region is causing the issue. They are probably already looking for better alternatives if they can find them.

CICS observability matters for resolving incidents, but it matters even more before issues occur. It realigns the organization's efforts around adding business value through innovative new features and better performance in front of customers and partners at lower cost, rather than expensive and morale-killing maintenance and firefighting exercises.



People that understand how to improve mainframe performance in the greater context of application performance will be in high demand—and mainframe-native observability solutions like C\Prof will help leaders get a handle on both.

About the Author

Jason "JE" English is Director & Principal Analyst at Intellyx. Drawing on expertise in designing, marketing and selling enterprise software and services, he is focused on covering how agile collaboration between customers, partners and employees accelerates innovation.

A writer and community builder with more than 25 years of experience in software dev/test, cloud computing, security, blockchain and supply chain companies, JE led marketing efforts for the development, testing and virtualization software company ITKO from its bootstrap startup days, through a successful acquisition by CA in 2011. He co-authored the book *Service Virtualization: Reality is Overrated* to capture the then-novel practice of test environment simulation for Agile development. Follow him on X at @bluefug.

About Rocket Software

Rocket Software is the global technology leader in modernization and partner of choice that empowers the world's leading businesses on their modernization journeys, spanning core systems to the cloud. Trusted by over 12,500 customers and 750 partners, and with more than 3,000 global employees, Rocket Software enables customers to maximize their data, applications, and infrastructure to deliver critical services that power our modern world. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located throughout North America, Europe, Asia, and Australia. Rocket Software is a portfolio company of Bain Capital Private Equity. Follow Rocket Software on LinkedIn and X or visit www.rocketsoftware.com.