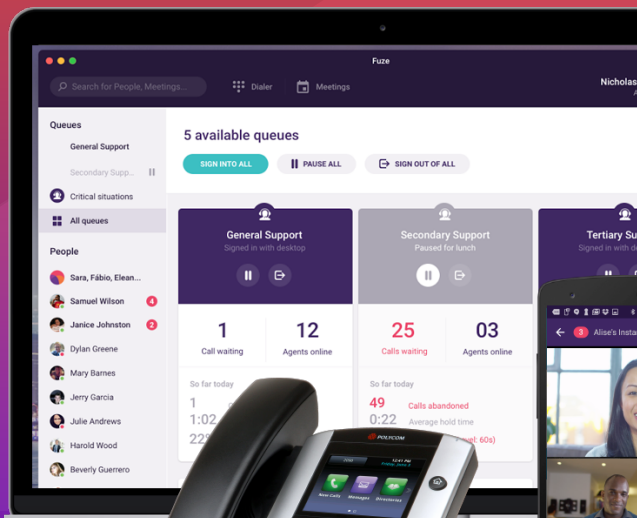
Customer story / Communication

How Fuze Improved Their Application's Heap and CPU Utilization by Over 50%

OverOps helped Fuze detect and fix critical issues faster, improving their core application performance and offering a better experience for their users



fuze

Fuze offers Unified Communications as a Service (UCaaS), and the team is responsible for the company's provisioning portal. This application manages most of the core business processes, such as quoting, contract management, provisioning services, managing handset and client configurations, invoicing and more.

The application is a combination of web-based UIs and REST APIs, providing customers, engineers and support teams with a variety of interfaces via which they can manage multiple operations. These operations include taking contracts and turning them into actual functioning services, along with building call flows, managing call centers and more.

#19 Forbes Cloud 100	700+ Employees	$300M+ Funding	UCaaS Visionary Gartner Magic Quadrant

Production Monitoring Ecosystem: OverOps splunk> Nagios

Highlights

- OverOps allows Fuze to find exceptions that don't show up anywhere else
- Thanks to OverOps, Fuze engineers see the performance impact exceptions have on their application
- With OverOps, Fuze reduced their heap size and CPU utilization by more than 50%
- OverOps has become a major part of Fuze's testing process, giving QA and engineers valuable information about each issue they come across

Tel: 1-800-324-3053 www.overops.com
343 Sansome St, San Francisco, CA 94107

Key challenges and pain points:

Fuze has been around for over a decade, and during those years the portal code base has exceeded over 12 million lines of code.
As the company evolved, some engineers moved between different teams, some started working on other parts of the application and, of course, new engineers are always joining us. At this point, the code base has become very extensive, with some portions that are very complex; and so the platform needs constant monitoring and handling.

Given that the portal is the central management application for our deployment, bugs can impact both our customers and internal stakeholders; and every issue becomes critical. Debugging is a significant part of our daily activities and detecting bugs, writing code fixes and deploying them to production can take anywhere from a few hours to a couple of days.

"OverOps was a big eye opener for us, giving us an in-depth look into the volume of each error and exception"

How has OverOps helped you solve errors and improve CPU Utilization?

One of our main focus points is application health. On one occasion we came across a specific exception that occurred over 3 million times an hour. Like most companies, our workflow used to consist of using log management tools to try and identify the frequency of the most common errors. However, those tools only gave out a partial image as to what's really going on within the application.

With OverOps, we are able to find every error or exception before it impacts our users. We are also able to see the impact of caught exceptions, and detect when their volume poses a performance risk. "OverOps helped us detect not only the issues we were aware of, but also helped us make our application run better for our users"

"OverOps helped us detect not only the issues we were aware of, but also helped us make our application run better for our users"

Thanks to OverOps, we were able to significantly reduce our heap size, improve CPU utilization and cut down our debugging time.

Within 2 months of deploying OverOps, we were able to improve CPU utilization and heap size by over 50%. A side effect of this was reducing the time it takes to run full automated regression tests to under two hours. Also, OverOps significantly helps our developers in finding the root cause of errors, and reduces the time it takes to actually fix them.

How are you integrating OverOps with your daily workflow?

Every single piece of code we write goes through code review, and our QA team have incorporated OverOps as part of their testing process.

"OverOps is an impressive eye opener, helping reduce heap size, improve CPU utilization and cut down debugging time"

Instead of searching through the log files trying to recreate a certain issue, they have OverOps to point them in the right direction.
The OverOps dashboard pinpoints where an issue happened within the code. When QA open a ticket, they add the relevant URL to the error's analysis from OverOps, giving our engineers a complete analysis of what actually happened.

Tel: 1-800-324-3053 www.overops.com
343 Sansome St, San Francisco, CA 94107

Full code and variable state to immediately reproduce any error.

No need to manually reproduce issues by searching for information in logs. Reduce MTTI by 90%+

<1% overhead in production

OverOps operates between the JVM and processor level enabling it to run in staging and production.

Proactive detection of all new and Critical errors

New issues are detected and routed to the right developer vs. discovered by users. Each error receives a unique code fingerprint unique it across the app.

No change to code or build

New issues are detected and routed to the right developer vs. discovered by users. Each error receives a unique code fingerprint unique it across the app.

Supported Platforms:

JDK 1.6 and above | HotSpot, OpenJDK, IBM JVM | Java, Scala, Clojure, Groovy | Linux, OS X, Windows | Docker, Chef, Puppet, Ansible | Coming soon: .NET

Integrations:

SLF4J, Log4j, Logback, Apache Commons Logging, Java Logger | Splunk, ELK, SumoLogic, and any other log management tool | AppDynamics, New Relic, Dynatrace | Workflow automation: Slack, HipChat, JIRA, Pagerduty | Webhooks | StatsD

Learn how OverOps can help you automate your deployments -
Schedule a demo with an OverOps monitoring engineer

Tel: 1-800-324-3053 www.overops.com
343 Sansome St, San Francisco, CA 94107