

## What is Plumbr?

Plumbr is a Java performance monitoring tool that automatically detects memory leaks, GC inefficiencies, and locked threads. Plumbr locates the specific line of code or error in configuration that is causing the problem in your application and equips you with a step-by-step solution for the fix.

## A tool for developers or operations teams?

Plumbr is being used by both, but with different goals:

- **Operations teams** use Plumbr in production environments to quickly fix performance threats before they affect end user experience.
- **Developers** use Plumbr to automatically detect performance bottlenecks in their code before they ship it to production.



Did you know that 47% of Java applications suffer from memory leaks?



Did you know that 8% of Java applications experience regular GC pauses of 10 or more seconds?



Did you know that 20% of Java applications repeatedly halt user transactions for more than 5 seconds because of locked threads?

## What performance issues does Plumbr detect?

### Memory Leaks

Plumbr detects the size of a memory leak and the rate at which it is growing, finds the root cause, and equips you with instructions to fix the incident.

A memory leak in Java is a data structure which cannot be garbage collected. If left unattended, a memory leak will keep growing and eventually kill your application.

### Garbage Collection Inefficiencies

Plumbr detects when GC is pausing the application threads for too long and recommends you to tune your GC configuration on your application behavior.

Garbage Collection (GC) is a process that reclaims memory by identifying and discarding unused objects. In order to do so, GC needs to stop the JVM every once in a while. The length of those stops is unpredictable and can exceed tens of seconds during which end users perceive the application as unresponsive.

### Locked Threads

Plumbr finds the root cause for the lock and tells you which locks get contended and end up locking up other threads.

Locking is a safety net present in all multi-threaded applications. Synchronizing access using locks is a way to guarantee thread consistency. However, poorly designed locks can and will degrade user experience, adding thousands of milliseconds to user transactions.

## How does Plumbr work?

Plumbr is based on analysis of more than 10,000 Java applications. Plumbr algorithms either raise an alert when you have a performance issue or confirm you the lack of problems when your Java application is working smoothly. All the analysis is conducted during runtime and for this Plumbr needs data from the actual application usage.

## How to start using Plumbr?

It only takes a few minutes to sign up and add Plumbr to your JVM. The whole process has been made as simple as possible.



Sign up at  
[www.plumbr.eu](http://www.plumbr.eu)



Download the  
Plumbr Agent



Attach it to  
your JVM



Begin receiving feedback  
of your applications'  
performance

## What makes Plumbr unique?

Plumbr is the only Java performance monitoring tool that is able to zoom in to the root cause of performance problems so thoroughly. It directs you to the exact lines of code or configuration causing the problem.

Plumbr contains an extensive knowledge base of known performance issues with third party libraries and more common programming mistakes that impact application performance. In many cases, this allows Plumbr not only tell you what causes a specific problem, but also explain the best way to fix it.

## Who is using Plumbr?

Plumbr offers value for small startups and big public companies from different industries. Here are some of our key customers.



Plumbr tool and your advice has saved us from a lot of sleepless nights.

**Aman Jain**

Java Software Architect, Adeptia

I was very impressed by your onboarding process. I got started in minutes.

**Carl Hasselskog**

Co-founder and CEO, Degoo Backup



[plumbr.eu](http://plumbr.eu)

[twitter.com/JavaPlumbr](https://twitter.com/JavaPlumbr)

[facebook.com/javaplumbr](https://facebook.com/javaplumbr)

**Plumbr**