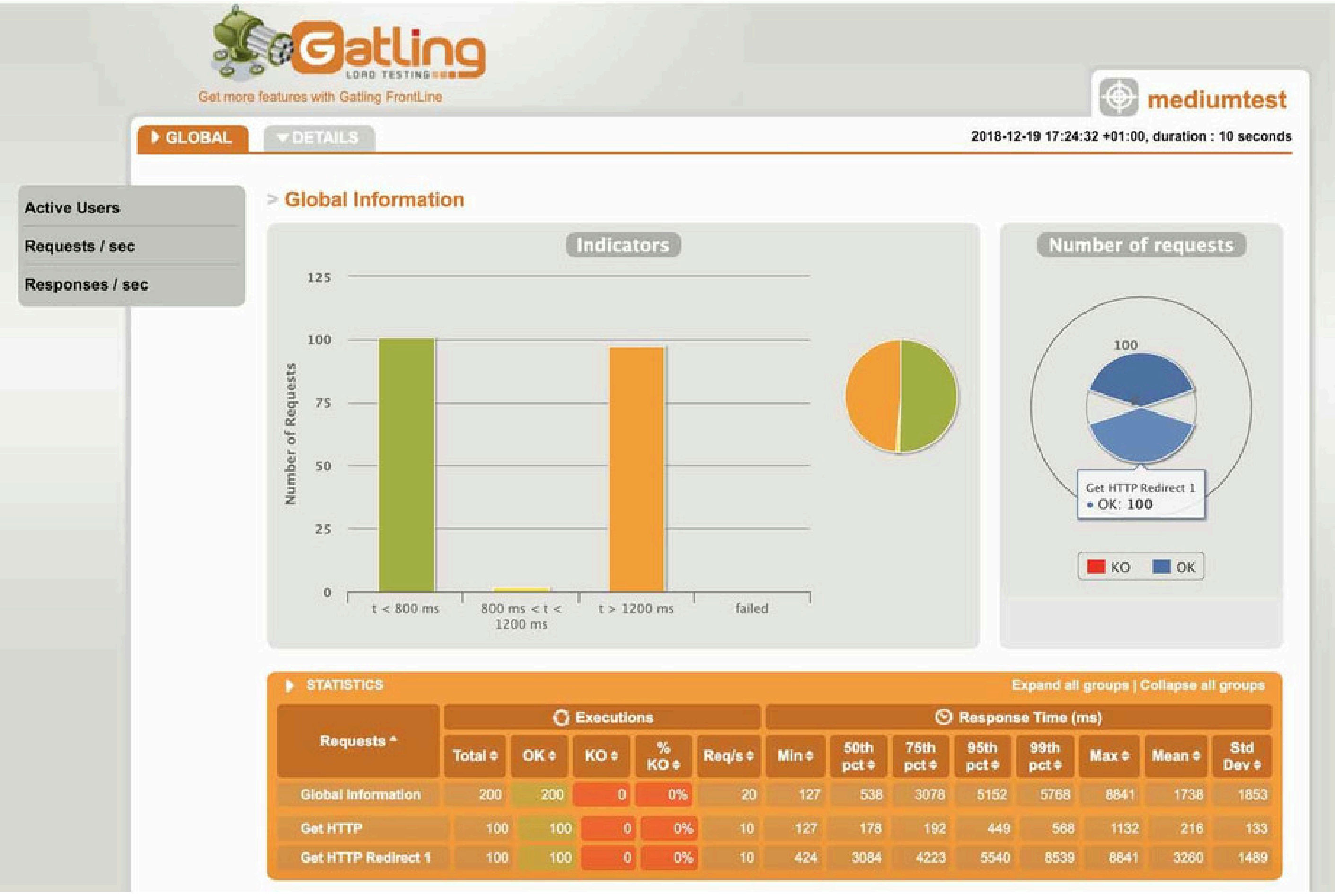
Reasons To Use Gatling For the Load Testing

Gatling is an open-source performance testing framework, which uses Scala, Akka and Netty as a technology stack and as its backbone. The first Gatling release was in January 2012. Since then, Gatling has had a couple of major releases almost each year, as well as a pretty extensive popularity growth across the performance engineering community.

As Gatling was created in order to write tests in the Scala programming language (which is a very powerful language but has a relatively small audience), you might think that this performance framework was designed mainly for a specific group of engineers or for limited needs. But if you check how many articles you can find about this framework in comparison to the other well-known open source load testing tools, you might be surprised, because its popularity stayed pretty strong over the past years.



Requests ▴	Executions					Response Time (ms)							
	Total ⬍	OK ⬍	KO ⬍	% KO ⬍	Req/s ⬍	Min ⬍	50th pct ⬍	75th pct ⬍	95th pct ⬍	99th pct ⬍	Max ⬍	Mean ⬍	Std Dev ⬍
Global Information	200	200	0	0%	20	127	538	3078	5152	5768	8841	1738	1853
Get HTTP	100	100	0	0%	10	127	178	192	449	568	1132	216	133
Get HTTP Redirect 1	100	100	0	0%	10	424	3084	4223	5540	8539	8841	3260	1489

Gatling Works Everywhere

Gatling is written in Scala, which allows you to run it on any system. That’s why you will never hit any trouble by using different local machines and cloud servers to run and create your tests.

Create Gatling Tests as Code

For many developers, it is a huge benefit if they can write performance tests as source code, as Gatling enables. This allows us to store tests under version control systems that enhance developers' collaboration, keep historical changes clear and prevent us from losing the work we have done. In addition to that, code helps bootstrap the tests by refactoring, which is a powerful process in any popular code IDE application where you can write the Gatling tests.

Gatling has Detailed Metrics Dashboards Out of the Box

Gatling created detailed metrics dashboard that you can see after tests execution without having to add any additional plugins. The report is stored as an HTML file, which can be easily saved for some future analyses and metrics comparison. In addition to that, the report is interactive. which allows you to perform more detailed analyses and concentrate specific requests in addition to the overall picture.

Gatling is Powerful!

First, it's worth saying that all performance tools should be powerful by the nature of their purpose. But Gatling definitely has an advantage over many other frameworks (like **JMeter**), because it uses advanced architecture and the Akka toolkit, which is based on the actor model that is distributed and fully asynchronous by design. Thanks to the Akka toolkit, Gatling doesn't allocate a separate thread for each user. Instead, all multithreading is done via messages between actors (universal primitives of concurrent computation), which allow to simulate a list of users by using just one thread.

Gatling has Capable Assertions

Gatling has an integrated assertions API, which gives you a full arsenal to perform functional assertions. This API allows you to run different types of functional checks along with your **performance testing**.

Gatling Uses Human Readable Tests

Gatling tests are very elegant, because Gatling defines a domain-specific language that allows writing very clear and human readable tests. This is important because it helps teammates work together on the same scenarios without spending time on additional knowledge transfer.

Gatling Provides Smooth Integration with Real Time Monitoring Tools

One of the most important aspects of performance testing is execution monitoring. Real-time monitoring allows you to keep execution under control. Also, sometimes it doesn’t make sense to wait until the test is finished as even initial results sometimes can give you all the answers.