Achieving Strength Through Chaos Engineering

If there's one performance measure you have to deliver on, it's uptime

eBook

KEYSIGHT

Introduction

It's the unexpected and unpredicted that will cause the problems – and in a world where systems are complex and inter-dependent, those are an accident waiting to happen.

The key to finding and fixing the unpredicted and unexpected is chaos engineering. This playbook looks at what chaos engineering is, how to conduct it, and the tools you need.

The difference between 99% and the gold standard of 99.9999% uptime is significant. It's the difference between over 3.5 days of downtime in a year, which would be unacceptable to many people, and just over 30 seconds, which is potentially barely noticeable.

The simple fact is that customers rely on your service and expect it to be available. It's bad for business – and your reputation – if you let them down.

Table 1. What does system uptime mean in terms of customer downtime?

System uptime percentage	Resulting customer downtime
90%	16.8 hours/week
99%	1.68 hours/week
99.9%	10.1 minutes/week
99.9999%	3.15 seconds/year

Examples of System Failures

- When Amazon's web services S3 system went down in March 2017, it was estimated to have cost U.S. businesses at least $310 million. [1]
- When the UK bank TSB experienced an online banking system failure in 2018, it cost the business £330m and lost it 80,000 customers. [2]
- When Facebook went down in March 2019 after a server configuration change, it attracted negative headlines across the globe. [3]

That's just the commercial sector. The systems that organizations in the aerospace, defense, and healthcare rely on are mission-critical. Downtime could literally put lives at risk.

The Difficulty in Maintaining Service

It's easy to say uptime is important. It's much harder to achieve in practice.

In the past, software ran in a highly controlled environment. Little could go wrong and problems were easy to find. Software updates happened very infrequently, allowing plenty of time for testing before release. [4]

Fast forward to today and we live in radically different times. Software runs on multiple servers and relies on distributed networks. Speed of release is a critical competitive differentiator but testing needs to be as robust and detailed as ever. [5]

In short, there are infinitely more opportunities for things to go wrong and it's infinitely harder to find the problem.

Testing goes some way to finding and fixing the problems. But by its very nature testing finds and fixes known problems or problems that can be anticipated.

It doesn't test for different configurations, different error conditions or the many factors beyond your control in an internet-distributed system, such as the failure of your third-party host server or a surge in usage. It's these problems that will really trip you up – and bring down your systems.

It is possible to find workarounds and fallback positions, of course – but you don't want to be finding them on-the-fly in the live environment. Instead, you need to have anticipated the problem and have a protocol to cope with it if it happened.

The key to finding the unexpected and unanticipated problems is chaos engineering – and it's anything but. It's the tightly controlled methodology developed by Netflix and used by enterprises worldwide who know that maximum uptime is central to their success.

The Concept of Chaos Engineering

The term 'chaos engineering' was invented by engineers at Netflix, but in many senses, the concept has always been with us.

Standard testing practices such as failure testing, disaster recovery and game days all look to achieve a similar thing as chaos engineering does – proactively perform experiments, inject failures and engineer disaster scenarios so solutions can be developed thoroughly and calmly rather than in the heat of the moment when the clock is ticking.

By conducting chaos engineering, developers and testers can have more confidence in the resilience of their systems in even worst-case scenarios – and according to the engineers at Netflix, it was a necessary evolution in the history of software development. [6]

Chaos engineering reflects a cultural shift within the software industry away from coordinated design and architecture, monolithic applications, and topdown engineering toward coordination of API boundaries, microservice architectures, and flattened engineering hierarchies.

As the complexity of these loosely coupled architectures increases, reliance on automated tooling to provide end-to-end tests for business-critical assumptions about the system becomes unavoidable. Confidence in the endto- end behavior of the system is manufactured by experimenting with worst-case failure scenarios in the production, scaled-out system."

Peter Alvaro, Kolton Andrus, Chris Sanden, Casey Rosenthal, Ali Basiri and Lorin Hochstein. [6]

How Does Chaos Engineering Differ from Testing?

But if much of what chaos engineering does is already present in standard testing setups, the question is: what makes it different?

Perhaps the biggest difference is that chaos engineering seeks to generate new information and find out what happens when something unlikely or unpredictable happens.

It is also true to say that traditional testing techniques typically have a binary output, making it the software equivalent of a closed question. Chaos engineering, in contrast, is the equivalent of an open question that reveals new areas for investigation.

What are the Benefits of Chaos Engineering?

There are benefits to all stakeholders from conducting chaos engineering and deliveringmore resilient and more reliable systems.

- Customers benefit because uptime is maximized. (Remember the gold standard of 99.9999% uptime and how that translates in practice.)
- Developers benefit because there are fewer critical incidents to solve and there are fewer damaging postmortems that affect team morale and a better standing within the organization.
- Businesses benefit from an improved reputation for reliability and lower levels of revenue loss caused by downtime.

So if the benefits of chaos engineering are clear, the next question is: how do you conduct it. We'll look at that in the next section.

Chaos engineering aims to create controlled chaos to see how much stress a system can withstand and to identify weaknesses. The website, principlesofchaos.org, defines the process of chaos engineering in four steps:

- Start by defining 'steady-state' as some measurable output of a system that indicates normal behavior.
- Hypothesize that this steady-state will continue in both the control group and the experimental group.
- Introduce variables that reflect real-world events like servers that crash, hard drives that malfunction, network connections that are severed, etc.
- Try to disprove the hypothesis by looking for a difference in the steady-state between the control group and the experimental group. [7]

As this definition shows, chaos engineering is far from chaotic. It is highly controlled and measurable. The idea is to understand what happens when chaos ensues – not to cause chaos.

There are also several best practices that have evolved to further add to the sense of control. These can be summarised as follows:

- Start small before scaling up.
- Start in a staging environment before moving to the production environment.
- Prioritize your activities and conduct the most important activities first.
- Calculate the cost of the experiment vs the ROI to understand its importance.
- Keep track of everything and be able to roll back to the steady state immediately.
- Don't conduct experiments on known problems – fix them first.
- Automate the testing when you understand a failure to maximize efficiency.
- Invest in Intelligent Exploratory Testing to increase the test coverage of your application, cutting edge technology powered by Artificial Intelligence will ease the testing load and increase productivity.
- Decide whether it's more effective and efficient to build your own tools to conduct the experiments or to buy tools. In either case, they must give you the ability to carry out best practice – i.e. intelligent, planned, controlled, safe, and secure experiments.

In the next section, we'll look at this last point in more detail and consider one of the tools available for conducting chaos engineering: Eggplant performance.

The Power of Eggplant Performance in Chaos Engineering

Eggplant performance provides open, extensible, and easy-to-use performance and load testing tools that can test the widest range of technology and can scale up to simulate any load you need. It offers testing teams six key benefits:

Application-level virtual users Eggplant performance is the only tool that allows testers to easily combine protocol-level client simulation to put the server under stress and verify the user experience at the same time. For example, it can place a load of 10,000 users on a server via protocol-level simulation and simultaneously validating the user experience on a mobile device, a tablet, and a PC (in several different browsers).

Scalable and distributable Eggplant Performance allows testers to partition virtual users any way they like across injectors. This means you can choose to run 100 virtual users from one injector, 200 from another injector, and so on. This means that Eggplant Performance is scalable to any number of virtual users. And by putting injectors in different geographical regions, it also means testers can analyze the impact of the client's location on the user experience.

Broad technical coverage and customizable virtual users Eggplant Performance has deep support for testing websites, but it also has in-built support for other common clientserver technologies such as Java, .NET, and Citrix. And for protocols and technologies that are not supported 'out-of-the-box', Eggplant Performance provides 'customizable virtual users'. This enables you to create new virtual user types that understand your protocols and technologies.

Dynamic run control Eggplant performance allows testers to modify the test parameters at runtime, for example, increasing the number of virtual users. Combined with real time monitoring of the server and client, this allows the testers to efficiently perform exploratory testing.

Data gathering, analysis, and presentation Eggplant Performance gathers comprehensive diagnostic information from the server and the client, and gives testers tools to analyze this information.

For example, statistical analysis of response times (including mean, max, min, and percentiles), ability to correlate factors, ability to compare test runs, ability to quickly aggregate and drill down into data, and the ability to import data from external monitoring tools and include this in analyses. Information about application errors is also gathered, exposing errors that might not normally be seen when the system isn't under load.

Easy but powerful automation script creation to create a typical Eggplant Performance automation script you simply execute the scenario once manually (e.g. by opening a browser and navigating the target website), and Eggplant Performance will record the network traffic and create a script. The system is able to handle the noise of factors such as sessions, cookies, dynamic forms, asynchronous communications, mobile devices, and security, which means that scripts are quick and easy to create and maintain.

Case Study: William Hill

Founded in 1934, William Hill has one of the most recognized, respected, and trusted brands in the gambling industry.

Founded in 1934, William Hill has one of the most recognized, respected, and trusted brands in the gambling industry. Employing more than 16,000 people in nine countries, the company continually works to transform its business, investing in new technology and innovation, and extending its vast product range to capture substantial growth and opportunities.

Sports betting is the largest and fastestgrowing section in the online gambling sector. Horse racing represents a significant proportion of the sports betting market. The Grand National race generates over £150 million, making it the busiest day for betting companies in the UK.

The problem: load

William Hill guarantees betting platform performance for the Grand National.

As the UK's leading bookmaker and one of the most recognized and trusted brands in the gambling industry, it is imperative that William Hill is able to handle the load on its online sports betting platform for the Grand National day.

The load placed on William Hill's systems during the Grand National period is significantly different than for any other race, and is comparable with the load increases seen during Boxing Day sales for online retailers.

The betting platform's infrastructure is hence significantly strengthened to support the workload, but performance testing this infrastructure is made more complex as it can only be conducted in live environments.

The goal

William Hill set the goal of maximizing the uptime for the company's web and mobile sports betting applications around the Grand National betting period. The company's performance testing solution is composed of Selenium WebDriver running on Grid with custom wrappers built to run from Amazon cloud instances. This testing solution was limited in terms of the accuracy of workload profiles, the repeatability and consistency of tests, and the scalability limitations of the tools.

> "The key to achieving our goal was being able to simulate as best as possible the transactions and user journeys that would be seen on the Grand National day.
>
> To achieve this we needed to build a robust performance test framework using a tool that was flexible, scalable, and capable of simulating large transactional volumes. And all of this with the ability to script using Java — our preferred language of choice. We also needed a performance testing tool that would allow us to build an accurate and repeatable process to undertake tests on our live infrastructure."
>
> **- SDET Manager, William Hill**

Eggplant Performance Selected

William Hill selected Eggplant performance as its new performance test solution based on several of the product's key features. Eggplant Performance provides sophisticated performance and load testing capabilities that can test the widest range of technology and can scale up to simulate any load, but are still easy to use.

The product helps ensure that applications continue to work, perform, and provide an excellent server performance when being accessed and used by thousands of simultaneous users.

And its script creation capabilities allow software development engineers in test to quickly create powerful scripts,Eggplant Performance also delivers significant productivity advantages.

The forecast transactional volumes were focused on William Hill's key business transactions and a general profile of browsing activity in the background.

The key business transactions consist of actions including bet placement, registration of new customers, and depositing funds from a card into an account. Testing coverage ranged from web apps on desktop and mobile devices like iPhone and iPad.

As the performance tests were undertaken in live environments, numerous control procedures were put in place to ensure there was no impact to customers.

Initially, about three tests were executed during the day, but as bottlenecks were resolved, tests became more complex and user numbers were tweaked for further increases in load.

Gradually, tests were executed on demand, and during the project new transactional records were set almost on a daily basis.

> "Ultimately, the testing load we generated using Eggplant Performance was more than the peak transactions compared to some of the world's busiest stock exchanges."
>
> **— SDET Manager, William Hill**

50 Performance Tests in 10 Weeks with Eggplant Performance

15 performance tests in 10 weeks last year.

In the build up to the 2015 Grand National, William Hill conducted a total of 50 performance tests over 10 weeks, compared to just 15 tests over the same time during the previous year.

On Grand National day, all of William Hill's betting systems coped with the workload and not a single high-severity incident was raised. In all, there were only two incidents for the entire day, and infrastructure remained stable.

> "The consistency of tests in Eggplant Performance helped us ensure that fixes applied between tests, were working as intended."
>
> **— SDET Manager, William Hill**

Conclusion

Today's developers are under more pressure than ever before. Uptime remains a significant performance measure but at the same time systems are becoming more complex and more dependent on third-party systems, making them harder to test.

Chaos engineering gives developers the methodology to find the problems and develop solutions out of the glare and before they happen. And one of the most powerful tools to support their efforts is Eggplant Performance.

At Eggplant we help businesses to test, monitor and analyze their end-to-end customer experience and continuously improve their business outcomes.

We provide business with award winning software such as the winner of Codie's Best DevOps tool 2019 – Eggplant Digital Automation Intelligence Suite.

Companies worldwide use Eggplant to surpass competitors, boost productivity, and deligh customers. How? By dramatically enhancing the quality, responsiveness, and performance of their software applications across different interfaces, platforms, browsers, and devices – including mobile, IoT, and desktop – in agile, DevOps, and innovative application and data environments.

We are a global company serving more than 650 enterprise customers in over 30 countries. Sectors include automotive, defense and aerospace, financial services, healthcare, media and entertainment, and retail. Eggplant is owned by Keysight Technologies (NYSE : KEYS).

KEYSIGHT

Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.