Challenges, Benefits and Best Practices of Performance Focused DevOps

Wolfgang Gottesheim
Dynatrace, Freistaedter Strasse 313, 4040 Linz, Austria
wolfgang.gottesheim@dynatrace.com

ABSTRACT

Did you know that just a handful of root causes are responsible for the majority of application issues like crashes, slow performance or incorrect application behavior? Non-optimized database access, deployment mistakes, memory leaks, or inefficient coding are just some examples. Companies that think Continuous Delivery and DevOps will solve all their problems typically fail as they just run into these problems faster. In this session we take a closer look at the most common problems, how to detect them and how to incorporate performance into your DevOps culture by automatically detecting these top problems.

1. DevOps – A DEFINITION

DevOps aligns business requirements with IT performance, with the goal of adopting practices that allow a quick flow of changes to a production environment, while maintaining a high level of stability, reliability, and performance in these systems. It is important to emphasize that DevOps is not about standards or tools, it is about enabling communication and collaboration between departments in an organization [1].

When talking about collaboration, a key aspect is how to prevent finger pointing between teams when problems occur. Performance issues are among the hardest to solve–they are heavily dependent on load, deployment, and user behavior, and operations (Ops) teams need help in identifying these issues and communicating them to development (Dev) in an actionable way.

2. MILESTONES TO PERFORMANCE FOCUSED DevOps

Tighten the Feedback Loops between Dev and Ops: Culture is the most important aspect because it changes the way how teams work together and share the responsibility for the end users of their application. It not only encourages the adoption of agile practices in operations work, it also allows developers to learn from real world Ops experiences and starts a mutual exchange that breaks down the walls between teams. From a performance perspective, it is important to establish a shared understanding of performance between Dev, Test and Ops. This enables collaboration based on well-known measurements and metrics, establishes a shared language understood by all teams, and allows all teams to focus on the actual problems. Finger pointing between teams has to be replaced by a practice that allows getting to the root cause of performance issues.

Establish a practice of automated performance monitoring: Ops and Test Teams usually have a good understanding of performance, and they need to educate developers on its importance in large-scale environments under heavy load. Providing automated mechanisms to monitoring performance in all environments, from continuous integration (CI) and test environments to the actual production deployment allows the shared language of performance to be spoken.

Measure key performance metrics in CI, Test and Ops: With performance aspects being covered in earlier testing stages, performance engineers on testing teams get time to focus on large-scale load tests in production-like environments. This helps to find data-driven, scalability, and 3rd party impacted performance problems. Close collaboration with Ops ensures that tests can be executed either in the production environment or in a staged environment that mirrors production, increasing confidence when releasing a new version.

Share tools and performance metrics across teams: More "traditional" testing teams are used to executing performance and scalability tests in their own environments at the end of a milestone. With less and less time for extensive testing, their test frameworks and environments have to become available to other teams to make performance tests a part of an automated testing practice in a CI environment. Automatic collection and analysis of performance metrics ensures that all performance aspects are covered by defining a set of performance metrics applied across all lifecycle phases, as this is beneficial to identify the root cause of performance issues in different environments.

3. CONCLUSION

The first step to move towards performance focused DevOps is to enable a shared understanding of performance through a set of key performance metrics that are accepted, understood and measured across all teams. These performance metrics allow all teams to talk about performance in the same way, and reduce the guesswork and finger pointing often associated with troubleshooting performance problems. Once these metrics are defined, their automated measurement and analysis is the next step that makes performance a part of a DevOps practice.

4. BIOGRAPHY

Wolfgang Gottesheim is a Technology Strategist in the Dynatrace Center of Excellence, he is involved in the strategic development of the Dynatrace Application Performance Management solutions. His focal points are Continuous Delivery and DevOps.

5. REFERENCES

[1] Kim, G., Behr, K., Spafford, G. 2013. The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. IT Revolution Press.

LT'15, February 1, 2015, Austin, TX, USA.
ACM 978-1-4503-3337-5/15/02.
http://dx.doi.org/10.1145/2693182.2693187