solarwinds

THE MISSED OPPORTUNITY FOR IMPROVED APPLICATION PERFORMANCE

Five DevOps Essentials for Better DBA and Developer Collaboration

By Thomas LaRock, Technical Evangelist and Head Geek

SOLARWINDS
WHITEPAPER

INTRODUCTION

For many software-as-a-service (SaaS) and Web-enabled businesses, database performance issues in production can cause a steady, relentless drain in business performance. For example, a self-service customer (or employee) portal fails to provide timely customer service because database lookups are taking too long. Or, frustrated agents begin logging an increasing number of complaints with the agent service desk because insurance quotes are timing out. Or, a customer transaction is repeatedly denied authorization, derailing a critical purchase and damaging a valuable business relationship. In fact, some estimate that as much as 70% of application performance problems are caused by problems with the database.

Performance issues in production can do more than damage a company's reputation and contribute to lost business. Fixing these issues in production is up to six times more costly than in development or testing. Problems in production can also delay full availability of the application by weeks or months, contribute to missing delivery windows and negatively impact customer expectations (and service levels).

The problems that plague application performance have much in common with the problems that have led many CIOs to adopt Agile development practices and DevOps approaches to streamlining service delivery. Agile has helped turn traditional waterfall development on its head, and enabled many businesses to achieve faster time to market by focusing on smaller, faster iterations with faster, more frequent releases to production and fewer defects. Efforts like ITIL, and more recently DevOps, have helped streamline operations, often with a specific focus on change and release management. Yet the same CIOs that have adopted these progressive approaches may be overlooking a critical opportunity: by applying the principles that underpin Agile and DevOps to application performance, they can create a proactive application performance approach that infuses the service delivery lifecycle from development to production. Proactively addressing performance, early in the development cycle rather than in production, pays off in many ways, including: reduced time to market, lower development cost and increased user acceptance of the delivered product.

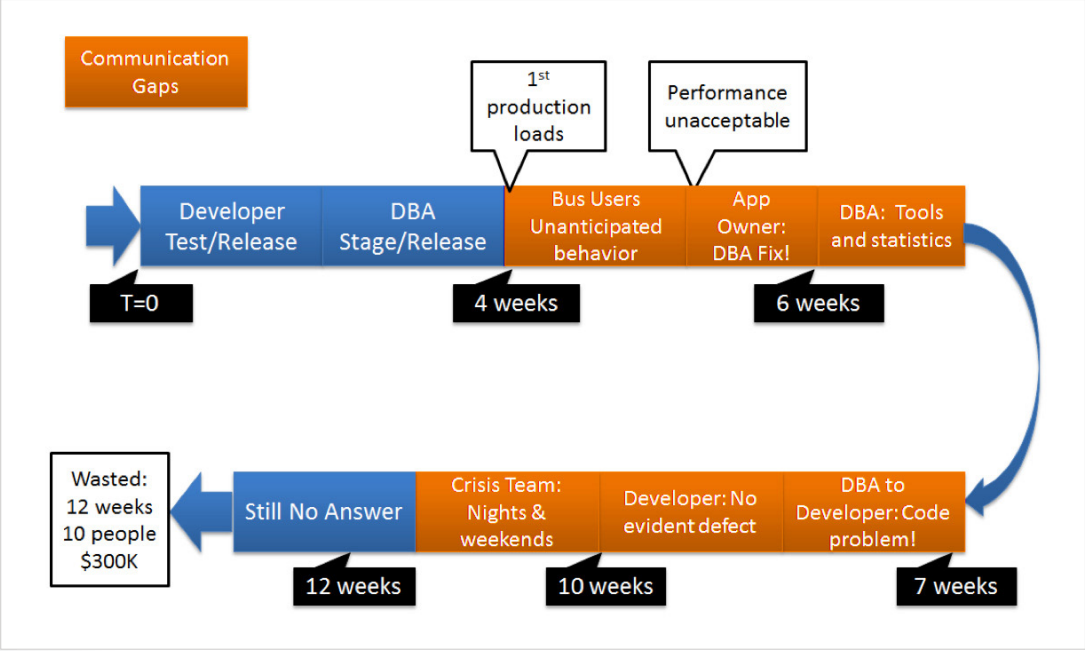
This paper describes specific actions that can be taken to build a proactive application and database performance monitoring practice that is shared throughout the development and IT organizations. Such a proactive approach serves to demonstrate the strategic value of IT in developing and delivering business-driving services, and contributes significantly to an organization's overall success.

THE ROOT OF APPLICATION PERFORMANCE PROBLEMS: SILOS IN DEVELOPMENT AND PRODUCTION

When application performance problems impact business, the finger-pointing between development and IT can reach a fever pitch, often resulting in no real resolution. DBAs, lacking insight to development, often blame the code. Developers, lacking insight to production, blame the database. Despite efforts made to solve the problem by pulling many different people into critical situation meetings, typically no one has a way to definitively identify the root cause of problems, and no way to rationalize priorities based on end user experience.

Underlying all these problems is the issue that, from environment to tools, the development and database production teams are extremely siloed and divided. Gartner estimates that, for 63% of large enterprises surveyed, the relationship between development and operations/production is somewhere between "uncollaborative" at best to "toxic" at worst[1]. This tension is very real, and can have very serious impact on the business

Consider this example from an insurance/financial services company, a composite drawn from our experiences with many organizations. Here, we track an application from development to release to a crisis situation:



- Development releases an application; performance has not been evaluated at all until this point.
- Four weeks later, after the application has experienced real production loads for the first time, the end business users start reporting problems; the application owner announces to all that performance is unacceptable.

- Six weeks on, the DBAs are striving to find the source of problem, using their own database tools.
- At seven weeks, after spending days or weeks fielding incoming calls from users and combing through mountains of reports and raw performance data, the DBAs conclude that because the database servers are running just fine, it must be the code, and they blame the developers.
- At 10 weeks, the developers confirm that the code, running in their development environment, is "fine" and conclude that the problem is the database.
- Having reached this toxic impasse, a crisis team is assembled, involving 10 people from multiple teams, and multiple meetings, often late at night or on the weekend.
- After 12 weeks, the crisis team still has no answer, and the organization has spent a minimum of $300,000 to arrive at that point (based on 10 people working on crisis management for 12 weeks).

In addition to these often hidden costs, the more visible costs of all this finger-pointing can have serious repercussions for the business, from being late to market to completely missing a market opportunity.

For organizations seeking to prove the strategic value of IT in delivering services, the case for a proactive approach to database performance management and monitoring could not be clearer.

FIVE KEY STEPS TO BUILDING PROACTIVE DATABASE PERFORMANCE MONITORING

At the core, a proactive database performance management approach requires communication, collaboration and integration across development and IT, with everyone focused on delivering exceptional end user experience. These are fundamentally the same principles that provide the foundation for many Agile, Lean and DevOps initiatives. While transforming an organization to embrace these principles is a journey that takes time, there are five key steps that can accelerate early successes:

- Provide developers direct monitoring visibility to test, staging and production servers
- Enhance collaboration by making developers self-sufficient problem-solvers
- Begin building performance into the development process by making it a functional requirement
- Establish shared metrics and a basis for equal access to metric reports across all teams
- Ensure that everyone in IT aligns with end-user expectations

Provide developers direct monitoring visibility to test, staging and production servers

Most developers lack the ability to monitor how their code is performing in the test, staging and production environments. The barriers can be significant. Most important, maintaining

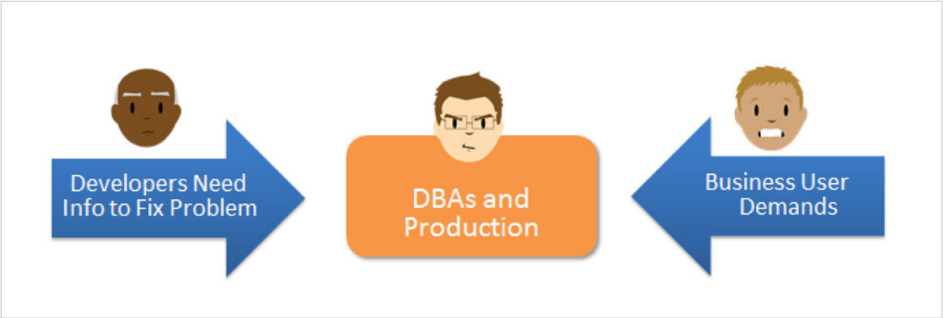
compliance with HIPAA, Sarbanes-Oxley and a growing number of other regulations, may mandate preventing access to production data. Other non-regulatory security measures may also lock developers out. Then, giving developers access to production data presents the possibility of loading production servers with more work. And, finally, if the divide between developers and DBAs in your organization is severe, there is likely to be a great deal of cultural resistance. However, the fact remains that if developers can't see performance of their applications in production, their ability to build code for production will be severely limited.

These barriers can, however, be overcome. Look for a monitoring and metrics tool that allows developers to see performance without exposing secure data. This tool should provide real-time and historical performance data, without putting extra load on critical production servers. Experience with customers has demonstrated that when given a tool with these capabilities, and the ability to see how the code performs in production, developers start making changes in development to reflect what they've learned.

Enhance collaboration by making developers self-sufficient problem-solvers

These barriers can, however, be overcome. Look for a monitoring and metrics tool that allows developers to see performance without exposing secure data. This tool should provide real-time and historical performance data, without putting extra load on critical production servers. Experience with customers has demonstrated that when given a tool with these capabilities, and the ability to see how the code performs in production, developers start making changes in development to reflect what they've learned.

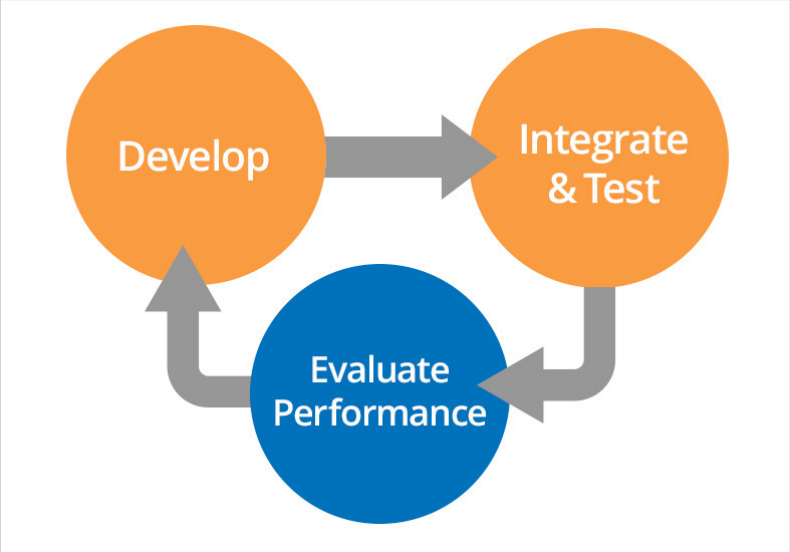
A further nuance on the need for production data is the need for data that is presented in a way that developers (and other non-DBAs) can truly understand. If developers have access to something like TRACE data, they may have some of the right information, but they are highly unlikely to have the level of expertise required to interpret that data in a meaningful way. Frequently, developers must ask DBAs to both pull reports and interpret them. This puts DBAs in the role of middleman between end users and developers, and can contribute significantly to communication issues.



To overcome this dilemma, look for a monitoring and metrics tool that gives developers performance data presented in a way that non-DBAs can understand on their own.

Begin building performance into the development process by making it a functional requirement

Look closely at how your organization addresses performance issues. Are most issues fixed well after development, as in our earlier example? If so, you must find ways to enable development and testing to evaluate production-level performance during development. Even better, find ways to integrate production into development, so that production requirements are understood from the beginning, by everyone, from architect to developer to QA to DBA.



To facilitate this process, look for a database performance monitoring tool that allows developers, QA and DBAs to work together throughout development and testing sprints to identify and address performance issues early, before they ever become production issues.

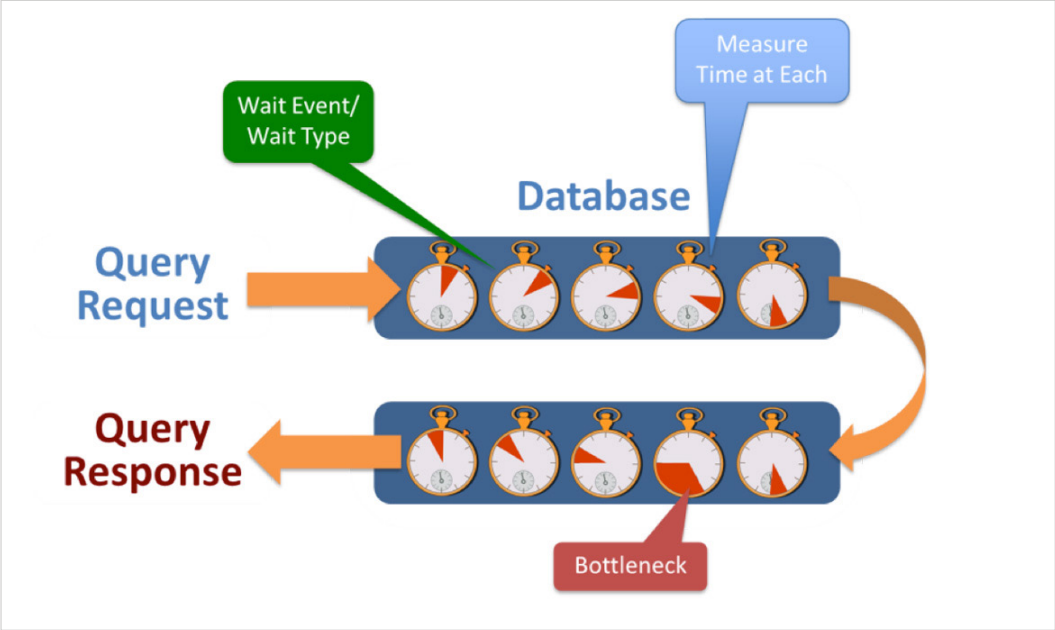
Establish shared metrics and a basis for equal access to metric reports across all teams

Development, production and management must all have a basis for sharing and comparing results, evaluating progress and tracking the real impact of changes made. If the metrics are to be understood by all, they must be simple both in concept and presentation. When the software and systems engineers have a common basis for discussion and progress evaluation, everyone is in better shape to work towards common goals.

Ensure all IT teams to align with end-user expectations

Finally, performance data that is presented in the context of the end-user experience is critical. If the performance data indicates that there is a higher than normal amount of contention one specific day each month, but there is no way to assess the impact to the end user, that information isn't very useful. The team might fix the underlying problem, but not see any correlation to end-user satisfaction. From developers to DBAs, everyone in IT must be able to hear, understand and respond to all database performance issues in the context of the end-user experience.

The most effective way to assess database performance relative to end-user impact is response-time analysis. Response-time analysis measures an end-to-end process, starting with a query request from an end user and ending with a query response to the end user, including the time spent at each discrete step in between.



Response-time analysis eliminates finger-pointing, enabling development and IT teams to work together in identifying specific database bottlenecks, pinpointing their root causes and prioritizing actions, all based on the impact to end users.

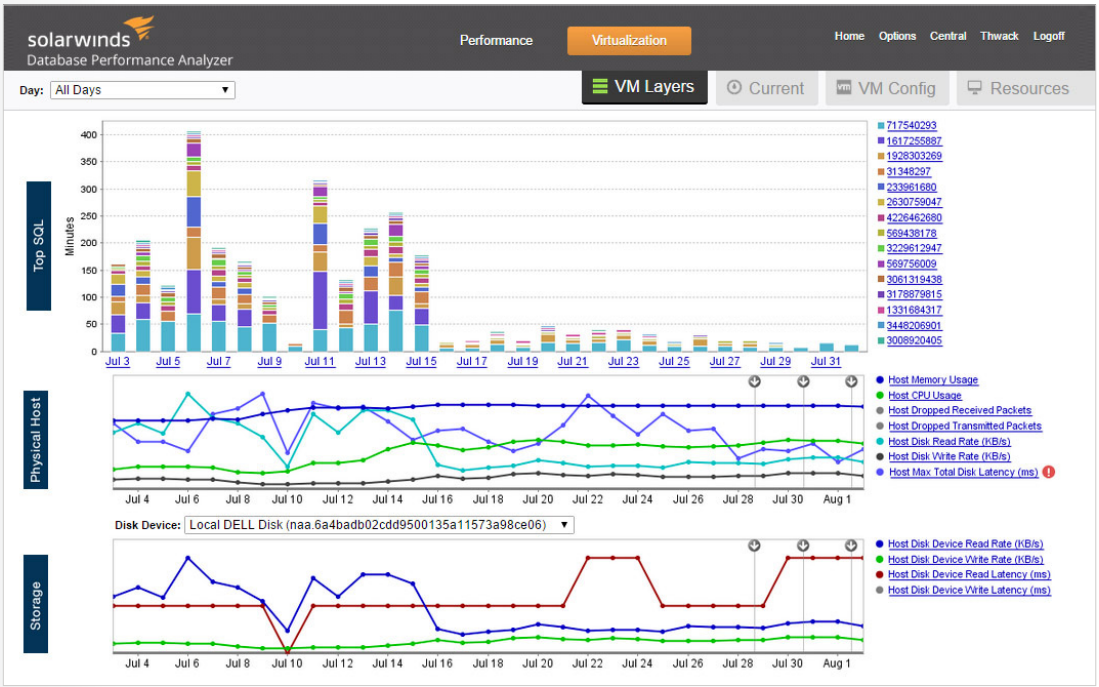
Though response-time analysis can be done with standard database monitoring tools, most require an advanced level of database management knowledge, and are beyond the reach of all but the DBA team. Instead, look for a tool that enables everyone from development to DBAs to do response-time analysis. The right tool can shorten and amplify feedback loops, so that everyone has the right information to quickly make corrections as needed.

Realizing the potential of proactive database performance management today

Many CIOs have embraced the need to reposition the IT organization as a strategic component of delivering world-class products and services. For CIOs in SaaS or Web-enabled businesses, an important element in this effort is the transformation from reactive to proactive application performance monitoring and management. Equipped with the right tool, organizations can begin the journey by building performance into development from the start and breaking down the silos that separate teams, from architects to developers to QA to DBAs to systems administrators.

How can Database Performance Analyzer Help?

Database Performance Analyzer (DPA) from SolarWinds (NYSE: SWI) provides the fastest way to identify and resolve database performance issues. DPA is part of the SolarWinds family of powerful and affordable IT solutions that eliminate the complexity in IT management software. DPA's unique Multi-dimensional Database Performance Analysis enables you to quickly get to the root of database problems that impact application performance with continuous monitoring of SQL Server, Oracle, SAP ASE and DB2 databases on physical, Cloud-based and VMware servers.



For additional information, please contact SolarWinds at 866.530.8100 or e-mail **sales@solarwinds.com.**

[1] "Catalysts Signal the Growth of DevOps," Cameron Haight, 8 February 2012, Gartner.