

OracleScene **additional**
DIGITAL **content**

Enterprise Performance Tuning in the Age of Agile Data

A typical DBA runs under tight constraints. Their resources are typically limited, their responsibilities are enormous and their praises rarely sung. As one DBA put it, “when I talk to my boss, it’s one of 3 Bs: beg, bitch or brag”.

Kyle Hailey 
Performance Architect
Delphix

The goal of this article is to get the DBA some huge bragging rights and have a direct impact on the revenue generated by their companies.

How can a DBA most improve the bottom line of the company they work for? DBAs in many organizations can have a massive impact on the goals of the IT department and thus the enterprise as a whole, not in the classic way of tuning a database but in a much more important way.

When it comes to improvements, a DBA usually thinks of database performance. The common ways for a DBA to improve performance is to modify database parameters, tune problem SQL queries or, in some rare but productive cases, actually get involved in designing the code that accesses the database. Everyone knows that code design is more important than simply tuning some database parameters or a few problem queries. Tuning queries or database parameters might improve overall performance by 10% while changing the application architecture might produce 10-fold gains (1000%). On the other hand, if one were to take a step back and say “how can I as a DBA make the biggest impact on my company’s bottom line?” one might be surprised to find out that re-architecting the application code access to the database, despite its huge payoffs, pales in comparison to another task that a DBA can perform. The DBA can eliminate one of the most common huge bottlenecks in application development.

What is the biggest bottleneck, the biggest constraint in application development? Application development projects are constantly running into delays and running over budget. A recent CIO magazine survey found that 60% of application projects were behind schedule. If we could eliminate the biggest constraint in these projects we could bring the projects to completion in time and under budget, not only creating huge savings but also allowing the company to start pulling in revenue from these applications.

.....
A recent CIO magazine survey found that 60% of application projects were behind schedule.
.....

When it comes to improving the throughput or flow of a system, the theory of constraints says that the only way to improve performance (throughput) of the system is to tune the bottleneck (alleviate the constraint). If one tunes anything other than the constraint it makes no difference on the overall throughput and, in some cases, makes throughput go down. If one improves performance ahead of the bottleneck then work will pile up at the bottleneck. If one improves performance after the bottleneck then those areas will be starved for work. Classic examples are adding more CPU

to a machine with an Oracle bottleneck that is constrained on something like latch contention. Adding more CPU power can actually make the bottleneck worse. Another example would be tuning a SQL query that no one uses any more.

With Oracle, it is easy to find the bottleneck in the system with wait interface instrumentation and monitoring interfaces such as Enterprise Manager's Top Activity page.

With Oracle, it is easy to find the bottleneck in the system with wait interface instrumentation and monitoring interfaces such as Enterprise Manager's Top Activity page. On another database without a wait interface or monitoring interfaces such as Oracle Enterprise Manager's Top Activity page, it is difficult to find the bottleneck. Since it is crucial to find the bottleneck to work efficiently, it is thus clear that it's important to have metrics that will identify the bottleneck.

When we look at application development, how do we find the biggest constraint? To clearly find the constraints requires metrics. What are the causes of developer lost days? How much time does it take for a developer to get an environment? How much of the QA cycle is spent building the QA environment versus actually QA'ing code? How much delay is there between the end of a sprint and when that code gets QA'ed. Unfortunately much of these types of metrics are missing in the industry. Fortunately we can look at the experience and analysis of industry experts to identify the most typical development projects delays. The bottleneck that is the most pervasive in the industry across IT departments is provisioning development and QA environments according to Gene Kim, author of The Phoenix Project.

Environment provisioning has made massive improvements in efficiencies over the past decade with the rise of virtual machine infrastructure allowing a machine to be spun up and configured from a template relatively quickly. What hasn't been addressed is the provisioning of data for these environments. Provisioning data,

especially large databases for QA and development environments, is the longest step in provisioning these environments. Provisioning data is where the DBA can come in and save organisations.

When polled, over 64% of DBAs say they use RMAN to create database copies.

How does a DBA typically provision a copy of a database to provide to development? When polled, over 64% of DBAs say they use RMAN to create database copies. How long does this take? The length of time it takes depends on the size and, more importantly, depends on the bureaucracy involved.

In talking to customers typical times are between a single day all the way up to 6 months. Why does it take so long? For one, the simple act of duplicating all the files in a database takes hours when the database is terabytes in size. The copying time can be eliminated with things such as storage snapshots. Storage snapshots only take seconds to make, but when we talk to customers using storage snapshots to copy database we still find that it can take days or months to get a database copy. Why does it take so long to get the database copy when the file copy only take seconds? It takes so long because there are so many hand offs between individuals and groups. The developer has to contact a manager to get a database copy. The manager contacts a DBA, who contacts a system admin, who contracts a storage admin etc.

The technology that solves all of these problems is the Database Virtualization Appliance (DVA).

One technology solves all these data mobility problems that cause delays in project development such as storage constraints, bureaucratic hand offs, slow provisioning times and lack of self-service. The technology that solves all of these problems is the Database Virtualization Appliance (DVA). A DVA is an innovative and powerful tool that is built to make data agile. A DVA takes the technologies

of storage snapshots and automates it to include collection of changes from the source database into the DVA. The DVA tracks changes from a source database, keeps those changes for a retention period and can provision a copy of the data from any point in that time window. The DVA manages compression, snapshotting and cloning of that data to provide automated provisioning of copies of that source database in a few minutes for almost no storage overhead. For example, two new clone copies of a database created by a DVA will initially share all their blocks. It isn't until the copies start modifying blocks that their storage foot print will start to diverge.

Where does one get a DVA? DVAs can be built by hand as some companies have done using 1000s of lines of code along with armies of Netapp filers. DVAs are starting to be offered by Oracle with limited functionality through Oracle's EM snap clone technology. A full DVA solution is offered by Delphix Corporation.

When a DVA is incorporated into a company's data strategy, it changes the way we think about data.

When a DVA is incorporated into a company's data strategy, it changes the way we think about data. With a DVA, data is handled in ways that were unimaginable or even impossible before. Instead of large teams of developers or quality control professionals sharing databases (and therefore creating bottlenecks on access and modification), each individual group or even team member can have their own private copy of the production database, no matter how big it is or where it is stored. Instead of waiting days or weeks for data, full-size read/write-capable databases can be provisioned in minutes. Instead of spending enormous (and ever-growing) amounts on storage and resources, companies are free to direct spending toward technical innovation and acceleration of application projects.

How big is the data tax that companies pay moving data around without a DVA? How much is the delay of database environment provisioning to development and QA impacting corporations? One way we can measure it is by looking at >>

Technology: Kyle Hailey

the improvements in project timelines at companies that have eliminated this data tax by implementing a DVA, transforming their entire enterprise data layer into an Agile Data Platform (ADP). Agile data is data that is delivered to the exact place it's required just in time and with much less time, cost and effort than copying data. IT experts building mission-critical systems for Fortune 500 companies have seen real project returns averaging 20-50% productivity increases after having implemented an ADP. That's a big data tax to pay without an ADP. The data tax is real, and once you understand how real it is, you realise how many of your key business

decisions and strategies are affected by the agility of the data in your applications.

Data agility is more than just a buzzword. It is a necessity. Thanks to the magic of virtualization, provisioning servers, storage, and software has never been easier. Public and private clouds can automate many of the requirements to build a feasible environment for development or production, sometimes in a matter of minutes. But when a system must store many gigabytes or terabytes of data, the data must still be copied into lifecycle support systems in order to bring a project to completion. The bottleneck

in application development is no longer the creation of servers, but rather the movement of the data required to provision environments and a DVA solves that problem. Development environments depend on having copies of production databases and a DVA allows provisioning in a few minutes with almost no storage overhead by sharing duplicate blocks among all the copies. Any DBA who can bring in a DVA to their IT department can radically impact the bottom line of their company's revenue by eliminating the enormous infrastructure, bureaucracy and time drag that it takes to provision databases for development environments. ■



ABOUT THE AUTHOR

Kyle Hailey

Performance Architect, Delphix

Kyle is a principal designer for the Oracle Enterprise Manager performance pages. He is a member of the OakTable Network, the co-author of Oracle Insights: Tales of the OakTable and was a technical editor of Oracle Wait Interface. He holds a patent in the area of database performance diagnosis and has been a speaker at Hotsos, NOCOUG, RMOUG, NYCOUG, Oracle OpenWorld and organises OakTable World. Kyle also teaches classes around the world on Oracle performance tuning. Currently Kyle works as a performance Architect at Delphix along with industry leading software, kernel and filesystem designers to take corporate data management to a new level of agility.

Get more out of your Oracle Applications

ORACLE E-BUSINESS SUITE | ORACLE FUSION FINANCIALS

APRO
SOFTWARE SOLUTIONS

From Order to Cash and Purchase to Pay...

APRO Software Solutions

www.arosoftwaresolutions.com

SOLUTIONS FOR:

- AP Scanning / Workflow
- Bank Statement Processing
- Payment / Direct Debit File Creation
- SEPA Mandates
- E-Remittances

KEY BENEFITS:

- Integrated in Oracle Applications
- Real Time Database Information
- Oracle Release Independent
- Fast Implementation

