White Paper

Database performance management

the importance of time-based analysis

A White Paper by Bloor Research
Author : Philip Howard
Publish date : February 2013

Confio Ignite and IgniteVM should be looked at closely by any company wanting to proactively monitor and manage database performance

Philip Howard

Executive summary

It can be argued that the best and perhaps only real true measure of performance for any computer system is how long it takes to respond to a user's request for information or, en masse, how long it takes to respond to all users' requests. Given that this is the case, and we would argue that it is logically obvious, then the only sensible way to monitor the performance of any particular system is to measure the amount of time it takes to respond to those requests.

When it comes to database performance monitoring, with which we are concerned in this paper, most performance monitoring tools still don't measure how long it takes to do things, but instead focus on counting the number of things that have to be done. A time-based approach to measuring performance is not only more intuitive but is also more effective, not just in monitoring performance per se, but also in identifying the root cause of any problems that may arise.

Specifically, this paper discusses the various issues that surround database performance monitoring and how we believe that they should be addressed. Having identified the characteristics that you should be looking for in such a system we then go on to describe a solution, Ignite from Confio Software, which meets these characteristics.

The problem

No large-scale computer system runs smoothly all of the time. In any enterprise environment there will always be times when there are bottlenecks or when applications perform inadequately so that service levels are not being met. The issue, therefore, is about discovering what has caused any such problem and then fixing it.

In theory, fixing a problem is not an issue. However, this is only true if you accurately and precisely know the root cause of this problem. If, for example, you have a badly written SQL statement then re-writing it is not difficult. What may be difficult, however, is pinpointing this particular SQL statement as the root cause of the problem. This white paper looks at just how you go about doing this, particularly with respect to the identification of database performance issues.

Three key elements for monitoring database performance

There are three elements to bear in mind when considering the metrics that you need for monitoring database performance. These are:

1. What do you measure?

2. How deeply do you measure it?

3. What context (if any) do you apply to your measurements?

We will consider each of these in turn.

What do you measure?

The traditional way of monitoring databases for performance problems is to monitor the things that are happening in the database. This is what the utilities provided by database vendors typically do and it is also the methodology that most third party products use. What this approach involves is counting all of the things that are happening in the database. For example, how many physical reads there have been within a specified time period, the number of writes, lock statistics and so on.

However, in isolation none of these metrics means anything. How do you know, for example, if 19.2 million reads is a lot? Or is it not very much? Clearly you could apply trend analysis to this information so that you can compare the number of reads in this time period with the number in other time periods. But how much does that help? Suppose that you know that you have had an unusually high number of reads; you still don't know how that has affected performance or, if it has, whether this is the biggest contributor to any drop in performance. Moreover, even if the number of disk reads is adversely affecting performance, you still don't know what has caused this increase in disk reads. The cause could be anything from another system that has developed a fault and which has failed over to this system (in other words, we simply have an unusual number of users and processes and there is no problem) to poorly written SQL.

So, simply measuring numbers of operations does not tell you enough. We would contend that a preferable approach is not to measure the number of operations but the time taken by these operations. This is logical: end-user performance is what we are trying to maximise and that performance is essentially a measure of the time taken from a user submitting a query to him or her getting a response back. In other words, performance for the end user is time-based. It makes sense, then, for performance monitoring of the database to be time-based.

A good analogy is to a (very long) ladder in which the rungs are unevenly spaced and the time taken to go from one rung to the next is proportional to the distance between them. Working out the total time, and where holdups occur, is a matter of measuring time spent for each step and then adding all the time up: simply counting the number of rungs gets you nowhere.

Note that the argument against time-based measures is made with respect to the burden that they put on the database (measuring time periods is, in fact, more complex than simply counting). However, provided that there is no undue overhead imposed on database performance, then we believe that a time-based approach is still the preferred method.

How deeply do you measure?

There is a short answer to this question: your metrics must go down to the lowest level that it is possible to go. The whole purpose of database performance monitoring is to determine the root cause of bottlenecks that impede performance. If you consolidate performance data into buckets and cannot drill down below those buckets for details, then you cannot get to the root cause of any problem that may lie below the level of the bucket.

Note that the benefits of this approach are not limited to root cause analysis per se. The corollary to really knowing the root cause of a problem is that 'finger-pointing', where developers blame the database and DBAs blame the developers whenever there is a problem, goes away—it is no longer a question of each side blaming the other, since it will now be clear precisely where any bottlenecks derive from—so staff can get on with the more productive task of fixing the issue.

What context do you apply to measurements?

Somewhat surprisingly, many tools do not apply any context to the performance data that they collect. They simply collate numbers of operations as we have already discussed. However, bear in mind, again, that the performance we are ultimately looking to optimise is the performance experienced by end users.

Three key elements for monitoring database performance

Given that that is the case, then it will make sense to measure performance in terms of user requests to the database. That is, what is the total turnaround time for each user request, from initiation to fulfilment? This, after all, is what service level agreements are all about.

Summary

Let us put all of this together. Ultimately, database performance is the summation of every end user's experience of that performance. Therefore you must measure every user's database requests, break each of these requests down into their smallest possible components, and then measure the amount of time that each of these smallest components takes. You should now be able to see which parts of each user request is taking the most time, prioritise your activities because you can see where the most significant delays are occurring, and conduct root cause analysis at the most granular level.

Further considerations for performance metrics

A further consideration that we have not mentioned is the ability to know how much you can impact a particular problem. Suppose that you have two major sources of bottleneck within a particular application. To what extent can you impact those performance bottlenecks? It would be useful to know (and this applies regardless of whether you are using a time-based or count-based approach) by what percentage you can improve a particular function. If you can improve the performance of one of your bottlenecks (say, poorly written SQL) by an estimated 25% and the other by only 8% then it will make sense to focus on the former rather than the latter. So it will be advantageous if your solution can provide this sort of estimating capability.

However, this is not all. First, as we have previously mentioned, any solution must only have a minimal impact on the database you are monitoring. Second, while estimates suggest that as much as 70 to 80% of performance problems derive from the database this is certainly not the only cause of bottlenecks, with potential problems occurring either within Application Servers or across the network. It will therefore be useful if the supplier of any database performance monitoring tool also provides comparable capabilities at these levels within the infrastructure and/or will integrate with third party products that do this.

Third, while on the subject of integration it will also be helpful if any solution integrates with more general purpose database management tools, whether provided by third-party suppliers or the database vendors themselves.

Finally, since most environments have multiple, heterogeneous databases in production it will be useful, though not essential, if a single product and installation process can be used to monitor all of the databases in production. This should also extend to the ability to monitor databases running on virtual machines.

Confio Ignite: a solution that enables true time-based analysis

So far in this paper we have described different approaches to database performance monitoring and suggested what we believe to be a preferred solution, namely a time-based, highly granular, user-oriented approach. Now we will consider such a solution in detail, namely Ignite from Confio Software (www.confio.com/performance/database/).

Agentless architecture

The architecture of Ignite is illustrated in Figure 1, and there are several significant points to note. First, Ignite supports each of the databases shown: SQL Server, Oracle (including Exadata), DB2 and Sybase (SAP) ASE. In fact, Ignite allows you to monitor all these database instances from its web-based user interface, making all this performance data accessible without an installed client.

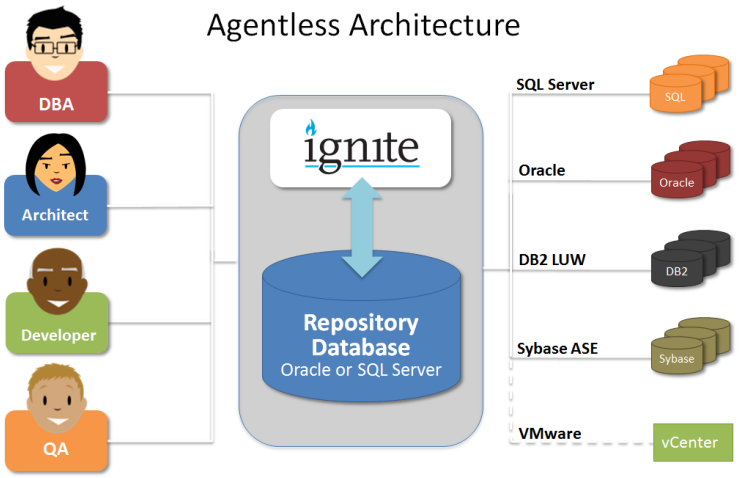

Figure 1: Performance detail on database performance with minimal load on monitored servers

Second, note the Ignite Repository Database. Ignite's architecture is truly agentless, and meets our stated requirement for a performance monitoring tool that does not put a load on the monitored databases. The Repository Database is where all the performance data is collected and the calculations of wait times are conducted and collated so that no overhead is imposed on the live systems for this purpose. In practice, Confio polls the live systems (reading information held in memory) on a regular basis so that no software is ever installed on the production systems being monitored. Confio estimates that this approach means that an overhead of less than 1% is imposed on these live systems. Note that the server on which the repository resides does not have to be dedicated to Ignite.

Third, Ignite is virtualisation-aware and supports all of the core databases mentioned, when running under VMware. That is, it provides a true picture of database performance on VMware virtual machines for these databases. IgniteVM exposes the multiple layers of the database and virtual machine architecture, including database instance, virtual server and physical host server resources. By knowing what changes in the virtual server impact CPU, memory, or I/O the software preserves the ability to measure wait times in context and enables proactive time-based performance analysis in a virtualised database environment.

Confio Ignite: a solution that enables true time-based analysis

Using Ignite: a simple, graphical interface

When a user request is sent to a database, the database processes that request by breaking it down into a number of discrete steps, and then executing these steps. Depending on the database the number of these steps may number many hundreds. What Ignite does is to measure the length of time that each of these steps takes for each user request. The time taken for a step to complete is referred to as a 'wait time'.

However, while this is what Ignite is doing under the covers, this is mostly hidden from the user. Indeed, as Ignite is essentially a visual tool we will discuss how it works primarily through a series of screenshots.

Figure 2 shows Ignite's main dashboard, which provides a high-level overview of performance trends across monitored database instances.

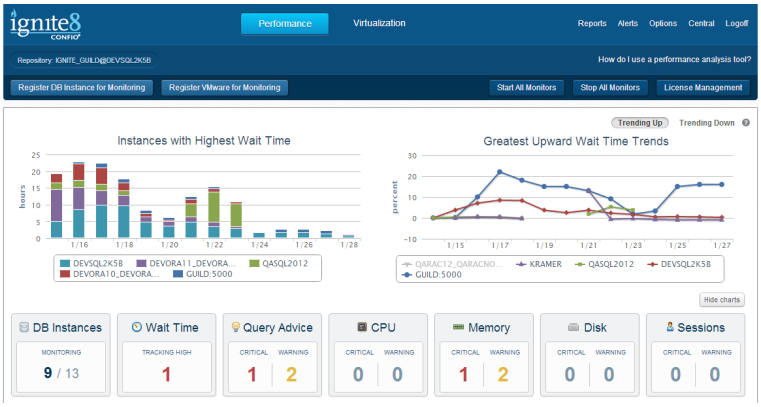

Figure 2: Ignite's dashboard focuses on top problems for all monitored databases

From this dashboard, you can start your investigations. Most of the options are self-explanatory. In Figure 3, below, we've drilled down to the Top SQL Statements report for a specific database instance, over a 24-hour period. In this view, the length of each bar indicates length of time.

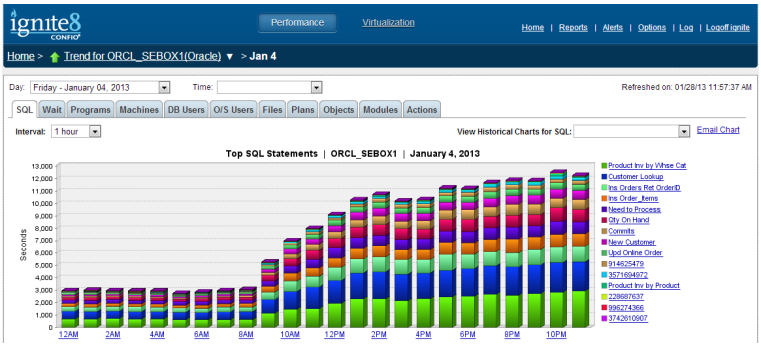

Figure 3: View of how database response changes through the day

Confio Ignite: a solution that enables true time-based analysis

Next we can drill down on a specific period of time to see what was happening during that time with much greater detail. In this view, as shown in Figure 4, Ignite ranks the specific query bottlenecks, also known as wait event or wait types, in the chosen period. The SQL queries with the most accumulated wait time across all sessions are charted to show the total delay for each.

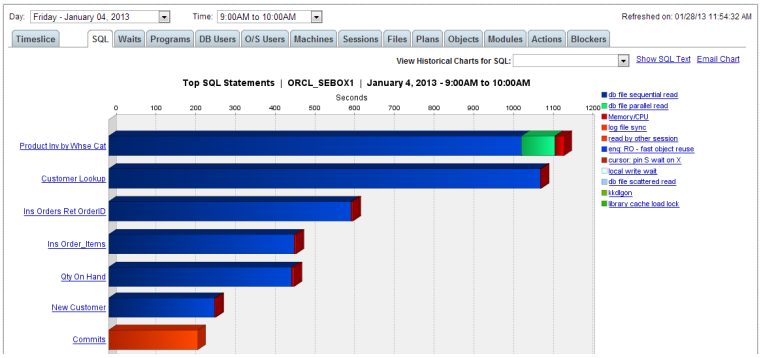

Figure 4: View of the correlation between SQL, response time and the specific wait event/type

As can be seen in this example, there is a problem with DB file sequential read for two particular SQL statements ("Production by Whse Cat" and "Customer Lookup").

What you would then do in this case is click on the SQL involved and, as can be seen in Figure 5, Ignite will display the actual SQL. Here, Ignite shows the specific SQL text and execution statistics, and you can use the tabs along the top row to drill down even further based on Programs, Sessions, Objects and more.

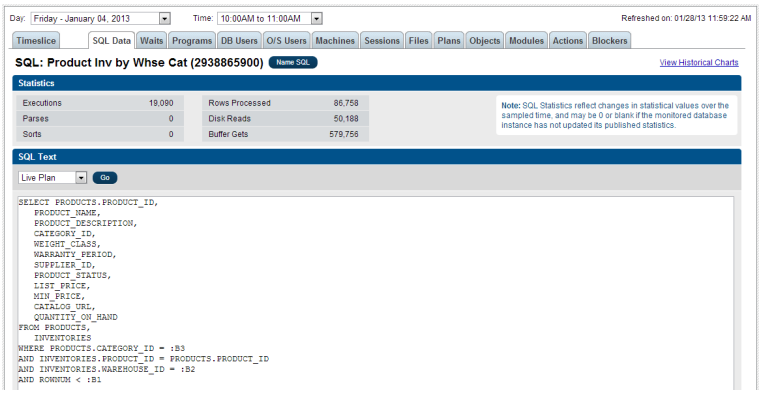

Figure 5: View of SQL-level detail

Confio Ignite: a solution that enables true time-based analysis

You can also roll-over the SQL bar and get a more detailed explanation of the wait event or wait type, and even get some hints for actions you might take. Now you have fully detailed visibility into the problem and the tools you need to discover the root cause of the problem.

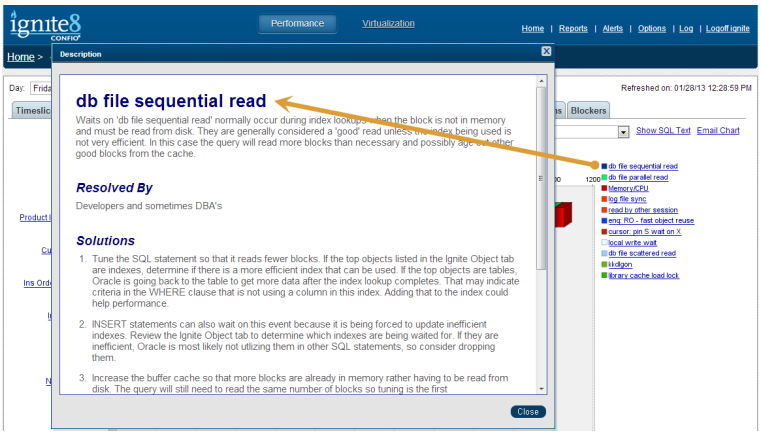

Figure 6: Built-in advice provides best practices on resolving issues

Ignite enables users to configure automatic and custom alerts, reports and metrics. Figure 7 below illustrates one such custom alert. These can be set up to send an SMS message or an email to a relevant party when a problem occurs or, more particularly, when a problem is about to occur.

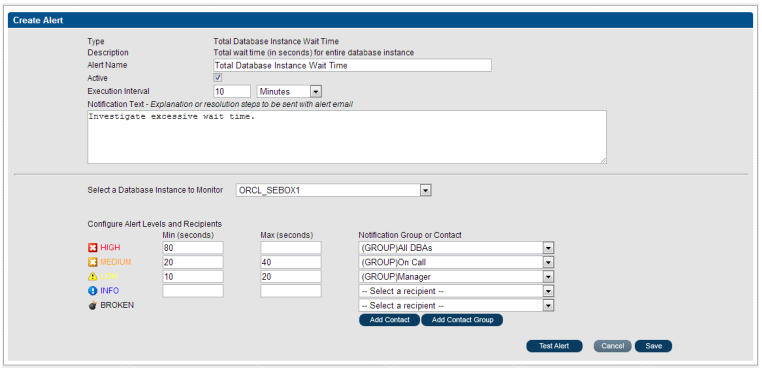

Figure 7: Setup Alerts to notify people before SLAs are exceeded

Finally, while we have so far discussed the examination of data in terms of wait times and SQL statements, it is also important to appreciate that you can also examine performance by user, which will be useful when a particular individual complains about poor performance. In addition, you can examine performance from a number of other perspectives including Programs, Machines, Sessions, Objects, Files, Plans, and Blockers.

Also, you can explore performance by application, although it should be borne in mind that this solution is primarily for database people who need visibility into what is happening at the application level, rather than application performance monitoring per se.

Confio Ignite: a solution that enables true time-based analysis

References

Ignite is a robust and proven database performance monitoring tool with thousands of users in virtually every industry. Many of Ignite's users have been prepared to publicly state their appreciation of the product and that it genuinely does do what it claims. We will not repeat all of these but interested readers can find details at http://www.confio.com/case-studies/. The comments range from endorsement of the ROI achieved to statements such as, "...within 24 hours we were able to pinpoint and solve a year-old problem."

Conclusion

The most important factor in any computer system is the people who use it. Performance is relevant only to those users and the way that you monitor any system (not just databases) needs to be aligned to users. The only realistic way to do this for database performance is to monitor the times that it takes to do the various things that need to be done. Moreover, you need to be able to do this in as granular a fashion as possible so that can you get down to the root cause of any problems. Confio Ignite and IgniteVM offer precisely such capabilities and should be looked at closely by any company wanting to proactively monitor and manage database performance.

Further Information

Further information about this subject is available from http://www.BloorResearch.com/update/2155

Bloor Research overview

Bloor Research is one of Europe's leading IT research, analysis and consultancy organisations. We explain how to bring greater Agility to corporate IT systems through the effective governance, management and leverage of Information. We have built a reputation for 'telling the right story' with independent, intelligent, well-articulated communications content and publications on all aspects of the ICT industry. We believe the objective of telling the right story is to:

- Describe the technology in context to its business value and the other systems and processes it interacts with.
- Understand how new and innovative technologies fit in with existing ICT investments.
- Look at the whole market and explain all the solutions available and how they can be more effectively evaluated.
- Filter "noise" and make it easier to find the additional information or news that supports both investment and implementation.
- Ensure all our content is available through the most appropriate channel.

Founded in 1989, we have spent over two decades distributing research and analysis to IT user and vendor organisations throughout the world via online subscriptions, tailored research services, events and consultancy projects. We are committed to turning our knowledge into business value for you.

About the author

Philip Howard
Research Director - Data Management

Philip started in the computer industry way back in 1973 and has variously worked as a systems analyst, programmer and salesperson, as well as in marketing and product management, for a variety of companies including GEC Marconi, GPT, Philips Data Systems, Raytheon and NCR.

After a quarter of a century of not being his own boss Philip set up his own company in 1992 and his first client was Bloor Research (then ButlerBloor), with Philip working for the company as an associate analyst. His relationship with Bloor Research has continued since that time and he is now Research Director focused on Data Management.

Data management refers to the management, movement, governance and storage of data and involves diverse technologies that include (but are not limited to) databases and data warehousing, data integration (including ETL, data migration and data federation), data quality, master data management, metadata management and log and event management. Philip also tracks spreadsheet management and complex event processing.

In addition to the numerous reports Philip has written on behalf of Bloor Research, Philip also contributes regularly to IT-Director.com and IT-Analysis.com and was previously editor of both "Application Development News" and "Operating System News" on behalf of Cambridge Market Intelligence (CMI). He has also contributed to various magazines and written a number of reports published by companies such as CMI and The Financial Times. Philip speaks regularly at conferences and other events throughout Europe and North America.

Away from work, Philip's primary leisure activities are canal boats, skiing, playing Bridge (at which he is a Life Master), dining out and walking Benji the dog.

Copyright & disclaimer

2nd Floor,
145–157 St John Street
LONDON,
EC1V 4PY, United Kingdom

Tel: +44 (0)207 043 9750
Fax: +44 (0)207 043 9748
Web: www.BloorResearch.com
email: info@BloorResearch.com