Manage performance and administer DB2 databases using IBM Data Server Manager

Overview and an example use case for performance management

Thuan Bui (tqbui@us.ibm.com)
Technical Enablement Specialist
IBM

Marcia Miskimen (mjmiski@us.ibm.com)
Senior Software Engineer QA
IBM

Somil Kulkarni (somilk@us.ibm.com)
Senior Development Manager
IBM

Shannon Rouiller (shannonr@us.ibm.com)
Information Architect
IBM

IBM Data Server Manager is a new, integrated tool for managing performance and administering DB2® for Linux, UNIX, and Windows databases. This tutorial starts with a brief introduction of Data Server Manager and a high-level explanation of the quick up-and-running process. The majority shows how Eric, a database administrator (DBA), can use Data Server Manager to investigate a statement performance alert. From start to finish, Eric uses one tool to learn about the alert, investigate the statement performance, review configuration changes, tune the statement, and review and implement the recommendations.

Overview

In this tutorial, we will touch on the simple setup experience for Data Server Manager and the extended feature capabilities, then we will walk through a specific example of investigating a performance alert, which will highlight several key features.
A solution for enterprise database management

To effectively manage DB2 for Linux, UNIX, and Windows databases across an enterprise, businesses can end up with a collection of disparate, narrow-scope tools for monitoring and administering those databases. Some of the most common feedback we hear from DBAs and IT staff:

- There are too many tools, too many installations, too many repositories.
- Tools need to be easier to use and need to support the team’s workflow, provide diagnostic smarts, and give actionable recommendations.
- Tools need to be more scalable and perform better.
- Tools must support managing databases across the enterprise.

Feedback like this provided the key requirements into the design and development of Data Server Manager which aims to be simple, scalable, and smart by being able to:

- Deliver a simplified user experience — from up-and-running to day-to-day use.
- Provide information via a single tool with a common integrated web console — not multiple tools with their own interfaces.
- Provide familiar capabilities from other InfoSphere® Optim™ database tools that enterprises depend on — capabilities such as performance monitoring and tuning, configuration, storage optimization, and database administration.
- Use expert analytics to help identify, diagnose, solve, and prevent performance problems.

Up and running with IBM Data Server Manager

IBM Data Server Manager is designed to get you up and running quickly. The setup program has only three steps, so you can use Data Server Manager within minutes.

Users and access

During setup, you specify a default login name for the Data Server Manager tool. This user ID is considered the default administrator for the tool, and it has all privileges to perform all functions within the tool. This user ID does not need to exist in the OS.

You can create additional users for login to Data Server Manager, and you can specify which databases they can work with. For direct access to a database itself, such as when running SQL or issuing other administrative commands to the database, users will still be required to provide their own DB2 credentials for that database. For example, you might have a group of DBAs who need only to view performance data but not to administer the database. This kind of separation of duties is fully supported in Data Server Manager.

Configuration

After you run the setup and log to in the web console, you need to define the databases you want to manage. Data Server Manager provides several ways to define these databases.
Add a database connection manually
You can define new database connections or modify existing connections at any time.

Discover the databases
Using DB2's discovery feature, the tool can find databases in your enterprise and add them automatically.

Import connections
The tool can import connections from a delimited text file that was exported from other InfoSphere Optim Database tools.

After a database connection is defined, it is immediately ready for use in the tool. From the left-side navigation menu, the two primary functional areas are:

Monitor
A default performance monitoring profile is applied, for monitoring and alerting out of the box.

Administer
A DBA can view and act on the database objects, edit and run SQL, create and manage database jobs, and view the database diagnostic logs.
Extending the capabilities of Data Server Manager

This repository is used to store historical data and for additional functions, such as advanced performance analytics, alerts based on deviations from historical states, and robust tuning and optimization capabilities.

Any supported DB2 for Linux, UNIX, and Windows database can be used as a repository, and it's a simple one-screen setup to make the switch. You have complete control over when to collect the monitoring data, how long to keep it, and which databases you want to collect it for.

Scenario: Solving a performance issue

Let's walk through some of the data monitoring, optimizing, and administration capabilities of Data Server Manager in a scenario.

Scenario:

Main flow of the performance issue scenario

1. View alert
2. Review access plan
3. Investigate statement
4. Track changes
5. Tune statement
6. Run DDL
7. Verify improvement

Eric is a DBA at the Sample Outdoor company. He's monitoring hundreds of databases for production, development, and test teams. He is using the full capabilities of Data Server Manager Enterprise Edition with a repository database.

One day, while monitoring the health of the databases in the enterprise, Eric notices that Data Server Manager has raised an alert for GSDBX, the database that is used by the development team for a new project. When the alert appears on the home page, Eric investigates the problem.

Viewing the alert

The Data Server Manager Home page shows a tile for each monitored database. Each tile shows some key metrics on gauges and an alert count indicator.

1. On the GSDBX tile, Eric notices the red (critical) alert count and a large increase in the I/O gauge. He clicks the alert count in red to see what happened.
The alert brief shows that a statement deviated from its past known performance.

2. Eric clicks the **View details** link to get more information about this statement alert. The alert details identify some specific metrics that deviated compared to past executions of this statement. Eric also notices that the Data Server Manager analytics suggest that a missing index could be contributing to the performance degradation.

3. Eric looks at the **Recommended Actions** tab for the alert. One of the possible suggested actions is to view the access plan.

**Reviewing the access plan**

Eric selects the **View access plan** link in the alert, which runs the Explain function and generates an access plan graph.
In the access plan graph, Eric notices a lot of table scans. He is surprised because he thought the tables that are involved in the query already have indexes. The high number of table scans is also consistent with the large increase in reads mentioned in the alert and with the increase in I/O that was shown on the Home tile gauge.

**Investigating the problem statement**

The alert also includes a link to tune the single statement, but before Eric does the tuning, he decides to look at some details about the historical performance of this statement.

1. Eric closes the alert dialog, clicks the **GSDBX** tile to open the monitoring details for the GSDBX database, and goes to the **Executed Statements** view. He sorts the statements by average rows read and sees the statement that triggered the alert.

2. Because Eric wants to see how this statement behaved in the past, he switches to the **Historical view** to analyze the data for the last three hours.
3. He drills down to look at the historical details for the interesting statement, and he sees that the performance really was better in the past. The blue bars show the statement execution time stacked with other metrics, but the darker blue for the statement execution time is the most prominent value. The red line is the execution count — which Eric sees was high while the execution time was low. When the execution time went up (the statements got slower), naturally the execution count dropped too.

```
Executed Statements: > select cust_cust_code, cust_first_name, cust_last_name, cust_total from gosalescust_sales where cust_cust_code = cust CUST TOTAL = (select max(cust_cust_code) from gosalescust_sales_header cust where cust_cust_code = cust CUST TOTAL and cust_order_date between ? and ?)
```

**Statement**

```
select cust_cust_code, cust_first_name, cust_last_name, cust_total from gosalescust_sales where cust_cust_code = cust CUST TOTAL = (select max(cust_cust_code) from gosalescust_sales_header cust where cust_cust_code = cust CUST TOTAL and cust_order_date between ? and ?)
```

**Time History**

<table>
<thead>
<tr>
<th>History time range:</th>
<th>2/24/2015, 6:33:00 PM to 2/24/2015, 9:33:00 PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total lock wait time:</td>
<td>0.000</td>
</tr>
<tr>
<td>Total statement execution time:</td>
<td>0:10:58.011</td>
</tr>
<tr>
<td>Total CPU time:</td>
<td>0:13:04.977</td>
</tr>
<tr>
<td>Total number of executions:</td>
<td>5.291</td>
</tr>
</tbody>
</table>

Eric recalls that the alert suggested the likelihood of a missing index, and he can see that the statement performed well before. Could this problem be a case of a dropped index?

**Tracking database changes**

Because Eric enabled the track changes feature for the GSDBX database, he knows that he can easily check whether any indexes were dropped:

1. Eric opens the **Track Changes** page to look at recent changes on the GSDBX database. He sees that three indexes were deleted.

Manage performance and administer DB2 databases using IBM Data Server Manager
2. He views details for the index objects and finds that the following three indexes were deleted: IDXONE, IDXTWO, and IDXONE_U. The timestamp is shown in GMT.

<table>
<thead>
<tr>
<th>#</th>
<th>Type</th>
<th>Time stamp</th>
<th>IND SCHEMA</th>
<th>IND NAME</th>
<th>CLUSTER FACTOR</th>
<th>CLUSTER RATIO</th>
<th>COL COUNT</th>
<th>COLLECTSTATISTICS</th>
<th>COL NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Delete</td>
<td>2015-02-25 02:52:00</td>
<td>GOSALESTC</td>
<td>IDXONE_U</td>
<td>-1.0</td>
<td>98</td>
<td>3</td>
<td>Y</td>
<td>&lt;cust_code,cust_last_name,cust_first_name&gt;</td>
</tr>
<tr>
<td>2</td>
<td>Delete</td>
<td>2015-02-25 02:52:00</td>
<td>GOSALESTC</td>
<td>IDXONE</td>
<td>-1.0</td>
<td>70</td>
<td>2</td>
<td>Y</td>
<td>&lt;cust_order_date,cust_total&gt;</td>
</tr>
<tr>
<td>3</td>
<td>Delete</td>
<td>2015-02-25 02:52:00</td>
<td>GOSALESTC</td>
<td>IDXTWO</td>
<td>-1.0</td>
<td>4</td>
<td>3</td>
<td>Y</td>
<td>&lt;cust_total,cust_order_date,cust_code&gt;</td>
</tr>
</tbody>
</table>

**Tuning the statement**

Now that he sees evidence of deleted indexes, Eric is ready to tune the statement to see if Data Server Manager has any new index recommendations:

1. Eric goes back to the alert he was looking at before, and opens the **Recommended Actions** tab again. The last time he opened the alert, he viewed the access plan and saw many table scans. Now he will choose the option to tune the statement.

2. Eric names his query, chooses to re-explain the statements, chooses to get recommendations for statistics and indexes, and clicks **Run**.

   Eric can go about his other work while the tuning job is running as a background job.
3. When the job is complete, Eric reviews the tuning results. The advisor has no recommendation for statistics because they are current. The advisor recommends one index, with a projected cost improvement of 99.79 percent. The left graph shows the estimated rows read by the statement without and with the recommended index based on the current statistics. The chart on the right gives an overview of the top operations by total cost; in this case, the operations will be all nested join operations using the recommended index.

4. Eric clicks the index recommendations number to view the details.

Eric notices that the recommended index matches one of the deleted indexes that he found earlier in the Track Changes report.

5. Eric decides to check the impact of the new index. Although Eric could run the Data Definition Language (DDL) right now, he wonders what impact this new index will have on the other queries running in the system — after all he tuned only that one statement. He clicks the Impact Analysis button to submit a job for the analysis.

The Impact Analysis job examines whether this new index will affect any other statements collected in the past. The tuning analytics engine will correlate these statements and determine if the originally recommended index would benefit other statements or if there is another index or set of optimal indexes for these related statements.

6. When the Index Impact Analysis job is complete, Eric views the results. The Original Index Recommendations section shows that this new index would impact three statements (two other statements in addition to the alerted statement.) In the Optimal Index Recommendations section, he sees a second index, and based on
the key columns, he notices that it matches another of the indexes that was deleted.

These two indexes also impact three statements.

7. Eric clicks **View Statements** to see what the other statements are, and he sees they are all projected to have significant performance gains if the new indexes are created.

In this example, the original and optimal recommendations have almost the same projected performance gain. Eric wonders whether it would be worth the extra resources to create that second index if the performance gain is only so slight. However, those other deleted indexes are still bothering him, and while he’s pleased and optimistic about the analysis results, he definitely wants to review the issue with his team.

8. Eric clicks the **Edit and Run DDL** button for the optimal indexes, to see the generated statements. He could execute the DDL to create the indexes right from the tuning job results screen, but he wants to review the DDL with his team and would need to create these indexes during off hours anyway to minimize the impact to the environment. Eric copies the recommended index DDL into his clipboard, and he cancels the window.

9. Next, Eric wants to save the DDL because he will need it later. Eric clicks **Administer > SQL Editor** from the navigation menu to open the SQL Editor and then he pastes the index DDL into the work area.

He clicks **Save** to save the DDL to a file on his local machine so that he can email it to his coworkers for the team review meeting.

### Running DDL from the SQL editor

At his staff meeting, Eric explains that he saw an alert about a statement that went bad, and that the track changes report identified three deleted indexes. The deleted indexes are very likely the reason that the statements are performing worse than before. The tuning wizard recommends
to create two indexes, which match those that were deleted. He has the new DDL all ready to
go for two of the indexes, but he wants to know if anyone knows what happened. One of Eric's
teammates realizes that he accidentally deleted all three indexes, but only meant to delete the
IDXONE_U index. Oops!

After some more discussion, Eric is now confident that he can re-create the other two dropped
indexes to get the database back to its intended state:

1. Eric logs back on to Data Server Manager web console, opens the SQL Editor and imports
the index DDL from the saved file into the work area. He modifies the recommended index
names to match the original names of the indexes that were deleted: IDXONE and IDXTWO.
2. After making his changes, he validates the syntax, just to make sure that there aren't any
mistakes.
3. Next, he clicks Run to execute the DDL statements. Data Server Manager
prompts for credentials to the GSDBX database, to be sure that Eric
is allowed to make such a change. The DDL is executed successfully.

Verifying the improvement

After the indexes are restored, Eric restarts the workload to confirm that the statements are again
performing well. He can verify this behavior in several ways:

1. From the Executed Statements view, Eric observes that the number of
rows read has decreased significantly because the index is being used.
2. Eric runs the visual explain again for the original statement identified in the alert. He sees the access plan graph now shows that the indexes are being used.

3. Eric looks at the Time History graph for the statement from the time before the indexes were dropped to the current time after they were re-created, and sees that the query is now performing more like its normal behavior.

After the workload runs a few times, Data Server Manager detects the improvement and closes the original performance alert.

Conclusion

In this tutorial, we showed an example scenario of how a database administrator can use Data Server Manager to efficiently monitor, analyze, identify, and solve a performance issue by using an integrated user interface without switching from tool to tool. In the process, we highlighted some key features of the tool, including smart alerts with solution guidance, performance tuning with
index impact analysis, configuration change tracking, performance analysis with historical view, and SQL Editor.
Resources

- Download **Base Edition** on developerWorks.
- Visit the **Data Server Manager Community** on developerWorks.
- Check out the **Data Server Manager video series**.
- Don't miss the **Data Server Manager documentation**.
- The **Information Management area on developerWorks** provides resources for architects, developers, and engineers.
- Stay current with **developer technical events and webcasts** focused on a variety of IBM products and IT industry topics.
- Follow **developerWorks on Twitter**.
- Watch **developerWorks demos** ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.
- Get involved in the **developerWorks Community**. Connect with other developerWorks users while you explore developer-driven blogs, forums, groups, and wikis.
About the authors

Thuan Bui

Thuan Bui has worked more than 25 years for IBM in development, quality assurance, customer support, performance, and management positions in database and data management technology. He currently works on the IBM Optim Solutions Technical Enablement team at Silicon Valley Lab, California.

Marcia Miskimen

Marcia Miskimen has more than 30 years' experience in the IT industry, with a background in all aspects of application and product development and support. She has worked with DB2 tooling for more than 10 years and is currently a senior software engineer on the quality engineering team for IBM Data Server Manager. She has a bachelor's degree in business administration/CIS from The Ohio State University, and has co-authored six IBM Redbooks publications.

Somil Kulkarni

Somil Kulkarni is a senior development manager of tooling for IBM Analytics Platform. He owns the development and program management aspects of IBM Data Server Manager (DSM) which brings performance, administration, tuning, storage and configuration on a common platform that is ready for the cloud and enterprise. He was instrumental in leading the development and architecture of the Optim Configuration Manager product suite (launched in 2011), the Q Replication product suite (launched in 2004), and tools like asnclp and asnpwd (launched in 2001). Somil has more than 15 years' experience in software development and leadership, and holds a master's degree in computer science from UCSB.

Shannon Rouiller

Shannon Rouiller is an information architect on the Optim Tools team in IBM Analytics. She has worked on the information experience for tools such as Optim Performance Manager, Optim Configuration Manager, and Data Studio. She has a bachelor's degree in math from Cal Poly and is one of the authors of Developing Quality Technical Information and Designing Effective Wizards.