

# WANT AN EASY WAY TO IMPROVE APPLICATION PERFORMANCE?



*Make a statement with attention to detail: Using cached SQL performance statistics to enhance application performance* – by Mike Behne (R&D Solutions Architect at BMC Software)

Optimizing the performance of SQL can have a huge positive impact for applications and system utilization. Do you know where to begin and how to make that impact? Db2 accounting traces are helpful to identify threads and packages that are consuming resources, but how do you narrow that down to the SQL statement level? For years, BMC has offered [APPTUNE for Db2](#) to make that accounting trace information visible at the statement and index/table level. With APPTUNE, you can quickly see the most expensive and longest-running SQL, with a wealth of supporting information to assist in understanding the tuning opportunities.

Even though analysis of metrics collected with this level of detail can provide opportunities for significant performance improvements, some users balk at the potential extra costs of collection. In cases where your system is well-tuned, or when you are looking for a way to identify an occasional problem, you need a way to provide this data efficiently on an ongoing basis.

## Cached Statistics

Over the past few releases of Db2, we have seen the addition of a new class of statistics that I'll call "cached SQL performance statistics". They first appeared uniquely for dynamic statements as a supplement to Dynamic Statement Cache (DSC) entries, established on first PREPARE and reused

for subsequent executions. These statistics are now also available for static statements as well, accumulating in the EDM pool supplementing the package skeleton content pulled from the catalog as established at Bind time. Although the statistics don't go to the same depth of detail as accounting records, they do provide useful information such as statement execution counts, elapsed/CPU timings, getpage & I/O activity, time spent waiting for locks, latches, I/O, and various other activities. Also unique to this class of statistics, you can see information related to access path execution, such as rows examined/processed, tablespace and index scans, and sorts performed, all helping to understand why the statement is performing the way it is.

## **IDUG North America 2019, June 2-6, 2019**

At the North America IDUG Db2 Tech Conference, come to session F12 (currently scheduled for Wednesday, June 5th, at 2:20 PM in the Carolina E room) and learn how you can capture cached SQL performance statistics, and see what's happening with your SQL database, at relatively low cost. We will run through some scenarios where poorly-performing SQL databases are identified and addressed with these cached statistics, showing how easy it is to find problem SQL and deal with it. I will also cover scenarios where more information may be needed to get to the bottom of a problem.

Come and be amazed at what is possible – and share your own stories of how you resolved performance problems.

If you are unable to attend, please visit BMCs website to learn more about [our solutions for managing the performance of SQL and Db2 applications](#).