

Modelo basado en aprendizaje de máquina estadístico para la determinación de factores que influyen en el rendimiento de sistemas de gestión de bases de datos relacionales

José Luis Ponce-Vergara

ponce_joseluis@yahoo.com / Universidad Nacional de Ingeniería

Recepción: 18/7/2020 Aceptación: 1/10/2020

RESUMEN. Los procesos de negocios que soportan sus operaciones con aplicaciones que interactúan con sistemas de gestión de bases de datos relacionales (RDBMS) pueden incrementar su productividad a través de la identificación de los factores que afectan el rendimiento de las ejecuciones de las sentencias SQL que conforman su carga de trabajo, especialmente las cargas generadas por aplicaciones implementadas en ambientes de producción, que son recurrentes en el tiempo. El artículo propone un modelo de identificación de factores que afectan el rendimiento de las ejecuciones de las sentencias SQL que se procesan en un RDBMS, valiéndose de algoritmos de aprendizaje de máquina estadístico (análisis de componentes principales y análisis de correlación canónica) que explotan la información de los planes, estadísticas y métricas generadas durante el ciclo de vida de las ejecuciones de las sentencias SQL.

PALABRAS CLAVE: SQL / sistema de gestión de base de datos relacional / análisis de componentes principales / análisis de correlación canónica / algoritmo de aprendizaje de máquina estadístico / rendimiento de ejecución de sentencias SQL / administración del rendimiento

A Model Based on Statistical Machine Learning for Determining Factors that Influence the Performance of Relational Database Management Systems

ABSTRACT. Business processes supporting their operations with applications that interact with relational database management systems (RDBMS) may increase their productivity through the identification of factors that affect the performance of SQL statement executions of any given workload, especially workloads generated by applications implemented in production environments which recur over time. This paper proposes a model to identify factors that affect the performance of SQL statement executions processed in RDBMS, using statistical machine learning algorithms (principal component analysis and canonical correlation analysis) that exploit the information of the plans, statistics and metrics generated during the life cycle of SQL statement executions.

KEYWORDS: SQL / relational database management system / principal component analysis / canonical correlation analysis / statistical machine learning algorithm / performance of SQL statements executions / performance management

1. INTRODUCCIÓN

La investigación se sitúa en el contexto de un área de conocimientos conocida como “administración del rendimiento” (*performance management*) y se enfoca en la construcción de un modelo que le ayude a un sistema de gestión de base de datos relacional con una carga de trabajo conocida o recurrente, como las que se manejan en ambientes de producción, a mantener un rendimiento aceptable de las operaciones que ejecuta a lo largo del tiempo, a través de la identificación de los factores que gobiernan ese rendimiento. Entiéndase el rendimiento como un requerimiento no funcional que un sistema de información debe cumplir para asegurar un adecuado nivel de servicio.

La investigación propone un modelo que se fundamenta principalmente sobre las investigaciones previas realizadas en dos áreas del conocimiento:

1. Reducción de la dimensión de un conjunto de datos para la identificación de los factores que influyen en el rendimiento de las ejecuciones de las sentencias SQL. El objetivo es reducir el número de variables de un conjunto de datos observados sin perder su esencia numérica (medidas de tendencia y varianza). El modelo propuesto utiliza el algoritmo de aprendizaje de máquina estadístico *principal component analysis* (PCA), al ser una técnica reconocida para estos menesteres, lo cual se fundamenta en la evaluación hecha por Mikolajczyk (2005). Asimismo, está probada su justificación y uso debido a la existencia de investigaciones que apoyan sus soluciones propuestas empleando el algoritmo PCA, entre las cuales se pueden citar a Badrinath Krishna (2015), Brauckhoff (2015), Shyu (2003) y Shawe-Taylor (2004).
2. Correlación no supervisada de variables para la identificación de relaciones de causalidad entre los factores que influyen en el rendimiento. Un algoritmo no supervisado supone que no se conoce *a priori* alguna relación de causalidad o dependencia entre dos conjuntos de datos. El modelo propuesto utiliza el algoritmo de aprendizaje de máquina estadístico no supervisado *canonical correlation analysis* (CCA), al ser una técnica reconocida en estos menesteres. El algoritmo CCA es usado para identificar y medir las asociaciones entre conjuntos de datos. Su justificación y utilización se puede consultar en las investigaciones realizadas por Zhang (2018) y Shawe-Taylor (2004).

Las investigaciones relativas al tema de análisis de rendimiento de *software* se pueden clasificar desde los siguientes tres enfoques: enfoque analítico, enfoque estadístico o de aprendizaje de máquina estadístico y enfoque de simulación.

- a. Enfoque analítico del análisis de rendimiento de *software*

Este enfoque normalmente se basa en teoría de colas, donde se modela al sistema de *software* como un conjunto de entidades servidoras dispuestas de diferentes maneras, las cuales tienen que procesar tareas en un orden de llegada preestablecido,

generalmente “Primero que entra, primero que sale” (*first in, first out*, FIFO). Algunas tareas tienen que esperar en las colas mientras otras están siendo atendidas por los servidores. La velocidad de atención de dichos servidores determinará el número de transacciones servidas por unidad de tiempo, así como la longitud de las colas de espera. Entre los trabajos que abordan este enfoque de manera extensa se mencionan a Kleinrock (1976) y Smith (2002).

- b. Enfoque estadístico o de aprendizaje de máquina estadístico del análisis de rendimiento de *software*

Este enfoque está basado en los datos (*data-driven*), y es donde se aplican algoritmos de aprendizaje estadístico o de aprendizaje de máquina. Este enfoque es el utilizado en este artículo. En este enfoque se aprovechan las técnicas de análisis de regresión estadística, econométricas o algoritmos de aprendizaje de máquina, con un conjunto de variables de entrada, y otro conjunto de datos con una o más variables de salida, las cuales se consideran dependientes de las entradas. Los conjuntos de observaciones para entrenar (*calibrating*) y probar (*testing*) los modelos son finitos, siendo el primero de mayor cardinalidad que el segundo. El modelo resultante es un modelo predictivo, entrenado y puesto a prueba con observaciones previas, y que permite estimar los valores de las variables de salida ante ciertos valores asignados a las variables de entrada. Entre los trabajos que abordan este enfoque de manera extensa se mencionan a Bontempi (2002) y Giusto (2001).

- c. Enfoque de simulación del análisis de rendimiento de *software*

Este enfoque está basado en la construcción de un modelo de simulación que captura las partes que marcan la cadencia del flujo de procesamiento del sistema de información que se estudia. El modelo de simulación imita el comportamiento en tiempo de ejecución del sistema, con el objetivo de ganar conocimiento de este e identificar sus cuellos de botella. Se puede consultar la investigación de Fortier (2003) para contar con una evaluación detallada de productos orientados con este enfoque.

Los modelos de simulación se construyen cuando la complejidad del sistema en estudio no permite usar enfoques analíticos conocidos para analizar y predecir su rendimiento. El modelo de simulación permite reducir el nivel de complejidad del sistema, y adecuar el nivel de detalle del modelo de acuerdo a la meta del análisis. Ejemplos de productos en esta categoría son:

- StateFlow de The Math Works: <https://www.mathworks.com/products/stateflow.html>
- COVERS de Analogic: <https://www.anylogic.com/resources/articles/covers-3-0-a-c-based-graphical-modeling-and-simulation-tool/>
- Anylogic de Analogic: <https://www.anylogic.com/features/>

1.1 Revisión de investigaciones previas en el campo de estudio

De *et al.* (2001) realizan una investigación empírica de factores influyentes sobre el rendimiento de consultas en bases de datos orientadas a objetos, que se concentra en la naturaleza de las consultas, sintaxis y semántica, así como los modelos de datos de implementación. Este tipo de enfoques requiere revisar cuestiones de diseño y programación de un sistema de información, tales como el código de las consultas y el tipo de modelo de datos elegido, lo cual impide escalar con facilidad la propuesta a otras situaciones.

Schkolnick y Tiberio (1985) plantean el desarrollo de fórmulas de costo para actualizaciones de bases de datos relacionales, como resultado de la evaluación de la sentencia a ejecutar, donde se considera como variables de entrada a los tipos de datos y predicados utilizados como filtros. Los autores proponen que este costo calculado influye considerablemente tanto en el diseño del modelo de datos como en las rutas de acceso a los datos.

Ganapathi *et al.* (2015) plantean un procedimiento para predecir métricas de rendimiento, donde se incluye al tiempo de respuesta de una sentencia SQL. Toma como entrada tanto los planes de ejecución de las sentencias SQL y sus resultados; de este modo, genera dos matrices: Query Plan Feature Matrix y Performance Feature Matrix. A partir de ellas crea proyecciones de rendimiento utilizando un motor de aprendizaje de máquina. La matriz de características del plan de la consulta (Query Plan Feature Matrix) es la que contiene los factores sintéticos que influyen en el rendimiento.

Como resultado de esta revisión, se observa que hay diferencias metodológicas al tratar de encontrar los factores de influencia en el rendimiento de las operaciones de un sistema de gestión de base de datos. Algunos enfoques analizan las sentencias SQL que conforman la carga de trabajo, y sobre este análisis generan conclusiones (enfoque de caja blanca), pero son más difíciles de generalizar por el trabajo que conllevan. Otros enfoques generan predicciones de rendimiento sobre la base de factores establecidos, y utilizan estos valores y los resultados obtenidos en las ejecuciones para entrenar y calibrar algoritmos de aprendizaje de máquina (enfoque de caja negra). La investigación que se presenta sigue el enfoque de caja negra, por permitir generalizar su metodología a diversas situaciones y obtener resultados en tiempos razonables.

El artículo presenta en la sección 2 la metodología que se sigue para generar el modelo de identificación de factores, luego en la sección 3 se muestran los resultados obtenidos por la aplicación del modelo en casuística real de ambientes de producción, y culmina con la sección 4 donde se presentan las conclusiones y trabajos futuros recomendados en esta línea de investigación.

2. METODOLOGÍA

La metodología empieza tomando como insumo las muestras de información relacionadas con la interpretación y ejecución de las sentencias SQL, clasificadas en dos grupos:

- Datos previos a la ejecución de las sentencias SQL: relacionados con el análisis e interpretación de las sentencias SQL. Son datos conformados por variables de tipo numérico contenidas en el historial de planes de ejecución de las sentencias SQL. Esta información es generada por el optimizador del motor de la base de datos antes de la ejecución de las sentencias SQL.
- Datos posteriores a la ejecución de las sentencias SQL: relacionados a la ejecución de las sentencias SQL. Son datos conformados por variables de tipo numérico contenidas en el historial de estadísticas y métricas obtenidas luego de la ejecución de las sentencias SQL.

Para dar inicio a la identificación de los factores que influyen en el rendimiento de las ejecuciones de las sentencias SQL, se seleccionan las variables numéricas representativas antes y después de la ejecución de las sentencias SQL. Las variables pertinentes al momento antes de la ejecución de las sentencias SQL se extraen desde el historial de los planes de ejecución generados por el optimizador de base de datos relacional y se agregan, debido a que el plan de ejecución de una sentencia SQL tiene la forma de árbol invertido. Las variables pertinentes al momento después de la ejecución de las sentencias SQL se identifican y extraen del registro histórico de las estadísticas y métricas de ejecución. Téngase en cuenta que tanto los planes de ejecución como las estadísticas y métricas son grabadas por el motor de base de datos relacional en un repositorio dedicado a ese fin, y dicha información es considerada como insumo del modelo propuesto.

Las variables obtenidas son reducidas dimensionalmente (simplificadas) a través de un algoritmo aprendizaje de máquina estadístico de reducción de dimensionalidad (análisis de componentes principales [PCA]), para, posteriormente, a través de otro algoritmo de aprendizaje de máquina estadístico (análisis de correlación canónico [CCA]) encontrar las correlaciones multivariantes de factores más representativos.

2.1 Entorno de ejecución y descripción de programas desarrollados

La metodología propuesta en el artículo se ejecutó en el siguiente entorno y considerando los siguientes recursos:

- Los resultados de rendimiento de ejecuciones de sentencias SQL corresponden a Bases de Datos Oracle Database 12c Release 1 Enterprise Edition (12.1.0.2) implementadas en entornos de producción. Estos resultados se graban automáticamente en un repositorio denominado Automatic Workload Repository (AWR).
- Las sentencias SQL que conforman las cargas de trabajo estudiadas incluyen principalmente sentencias DML (INSERT, UPDATE, DELETE) y consultas (SELECT).

- Herramientas ETL para adecuar los datos registrados en AWR al propósito buscado. La información de rendimiento de las ejecuciones de sentencias SQL es filtrada, procesada, agregada y cargada a un modelo de datos, para luego ser explotada por los algoritmos de aprendizaje de máquina estadístico basados en PCA y CCA. Estas herramientas se programaron a través de shell scripts y SQL scripts.
- Los algoritmos de aprendizaje de máquina estadístico se programaron en lenguaje R versión 3.6.1 para plataforma Windows 64 bits.
- Los tipos de bases de datos evaluados son listados en la tabla 1.

Tabla 1
Bases de datos de ambientes de producción en estudio

| Base de datos | Propósito (carga de trabajo) | Tipo de base de datos |
|---------------|-----------------------------------|-----------------------|
| 1 | Operaciones (transaccional) | OLTP |
| 2 | Gestión financiera (Batch + DSS) | DSS |
| 3 | MRP (transaccional y Batch + DSS) | OLTP + DSS |
| 4 | ERP (transaccional y Batch + DSS) | OLTP + DSS |

Elaboración propia

El proceso general se puede observar en la figura 1.

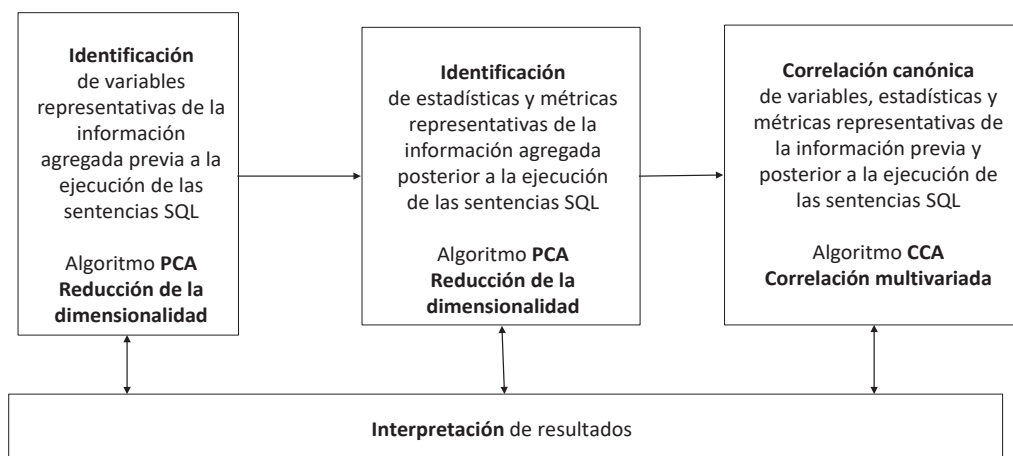


Figura 1. Proceso global de síntesis (reducción de dimensionalidad) y correlación canónica de variables que afectan el rendimiento de la ejecución de sentencias SQL

Elaboración propia

En la figura 2 se muestra el proceso de reducción de la dimensionalidad del conjunto agregado de datos previos a la ejecución de las sentencias SQL utilizando un algoritmo basado en PCA.

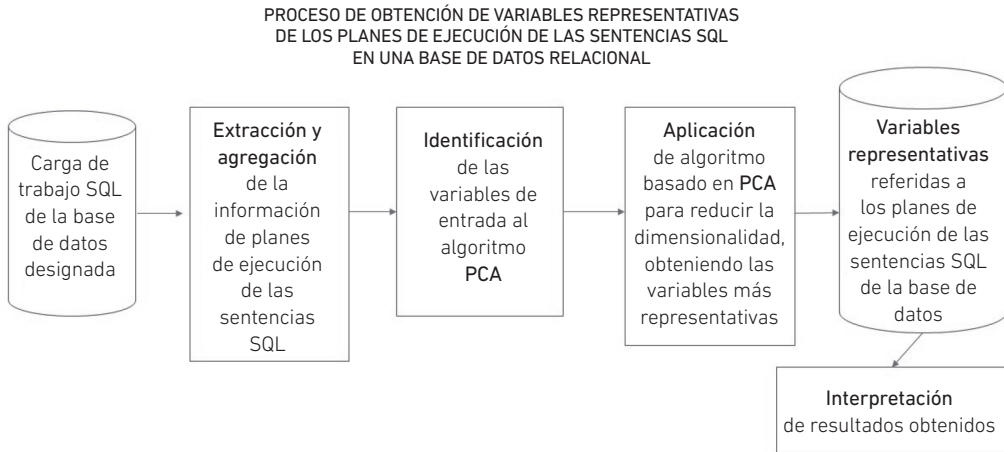


Figura 2. Reducción de la dimensionalidad del conjunto agregado de datos de planes de ejecución
Elaboración propia

En la figura 3 se muestra el proceso de reducción de la dimensionalidad del conjunto de datos posteriores a la ejecución de las sentencias SQL utilizando un algoritmo basado en PCA.

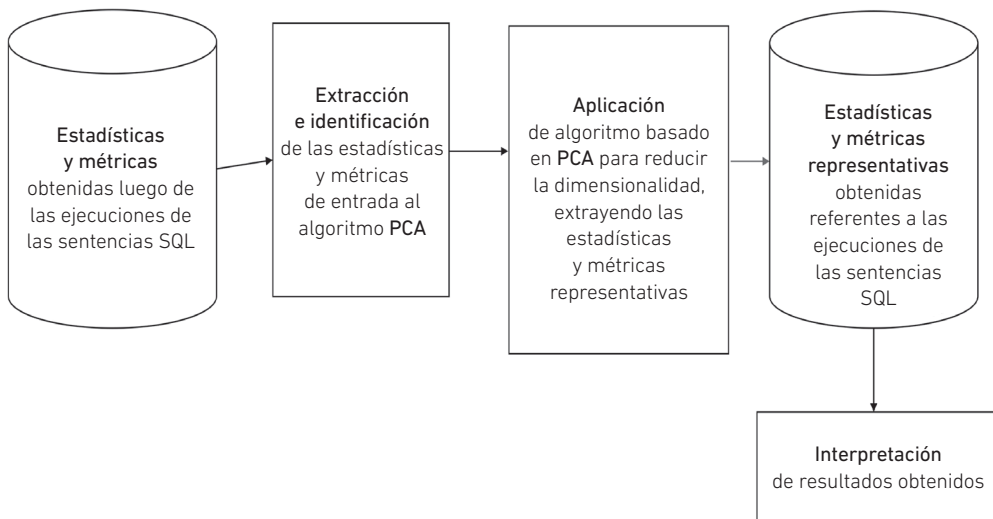


Figura 3. Reducción de la dimensionalidad del conjunto agregado de datos referidos al historial estadístico de ejecuciones
Elaboración propia

En la figura 4 se muestra el proceso que comprende el análisis de correlación canónica multivariante de los dos conjuntos reducidos dimensionalmente y la identificación de factores y relaciones que afectan el rendimiento del procesamiento de sentencias SQL.

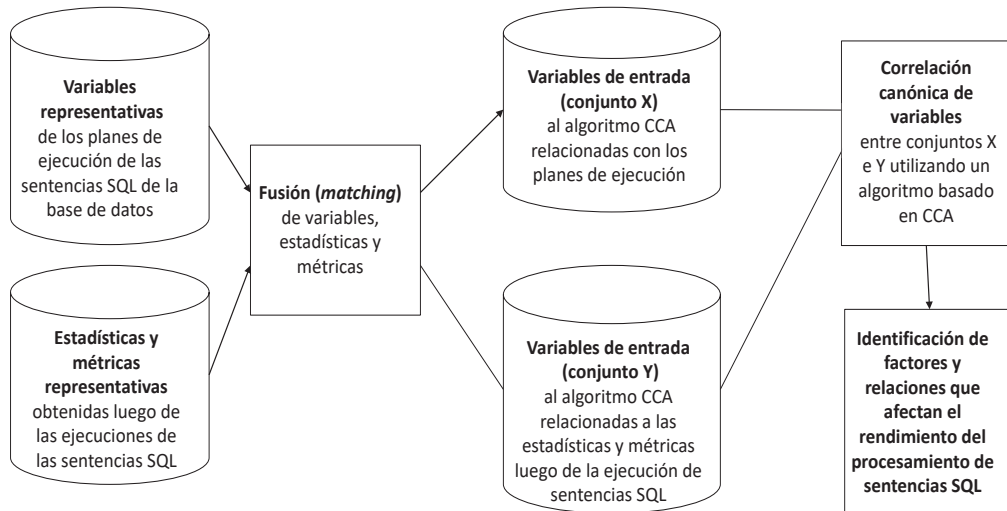


Figura 4. Proceso de análisis de correlación canónica multivariante
Elaboración propia

3. RESULTADOS

3.1 Algoritmo PCA a información previa a la ejecución de sentencias SQL

Tomando en cuenta la casuística de actuación de bases de datos de producción que sirven a aplicaciones DSS, y aplicando los algoritmos de reducción de dimensionalidad al conjunto agregado de datos previos a la ejecución de las sentencias SQL, se identifican los siguientes factores en el componente principal 1 (PC1), donde se concentra la mayor parte de varianza del conjunto de datos: I/O Cost, Cost, CPU Cost, Bytes y Cardinality (ver figura 5).

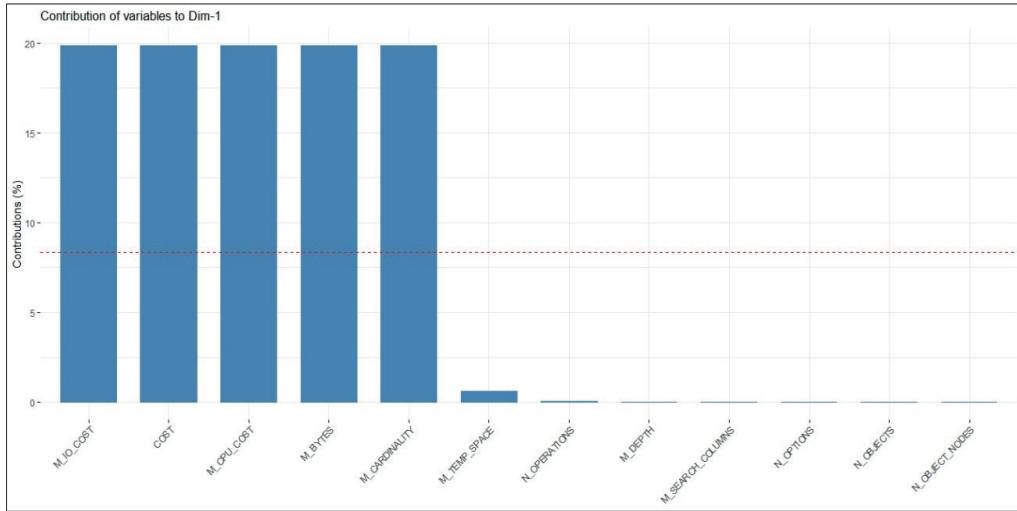


Figura 5. Variables representativas PCA1 de planes de ejecución para base de datos DSS
Elaboración propia

Para bases de datos OLTP, se identifican los siguientes factores en el componente principal 1 (PC1), donde se concentra la mayor parte de varianza del conjunto de datos: I/O Cost, Cost, CPU Cost, Bytes y Cardinality (ver figura 6).

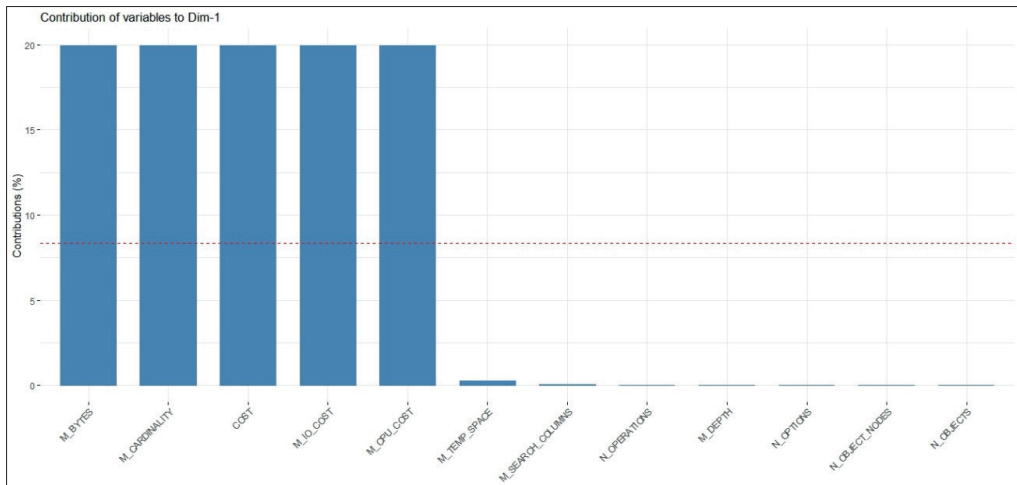


Figura 6. Variables representativas PCA1 de planes de ejecución para base de datos OLTP
Elaboración propia

Para bases de datos con carga de trabajo mixta (OLTP y DSS), se identifican los siguientes factores en el componente principal 1 (PC1), donde se concentra la mayor parte de varianza del conjunto de datos: I/O Cost, Cost, Temp Space, Bytes y CPU Cost (ver figura 7).

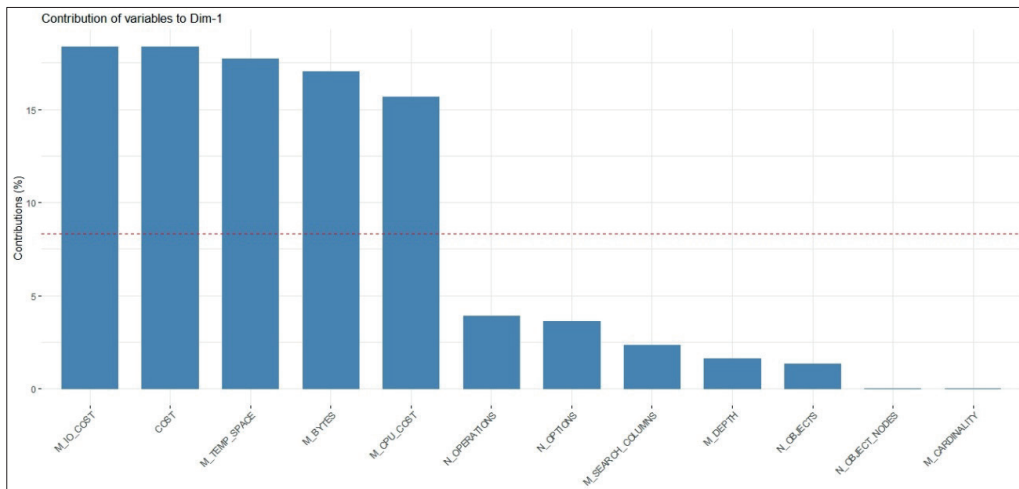


Figura 7. Variables representativas PCA1 de planes de ejecución para base de datos con carga de trabajo mixta OLTP y DSS
Elaboración propia

3.2 Algoritmo PCA a información posterior a la ejecución de sentencias SQL

Aplicando los algoritmos de reducción de dimensionalidad al conjunto agregado de datos posteriores a la ejecución de las sentencias SQL, para una carga de trabajo DSS se identifican los siguientes factores en el componente principal 1 (PC1), donde se concentra la mayor parte de varianza del conjunto de datos: Elapsed Time, I/O Wait, Disk Reads, I/O Interconnect Bytes, Physical Read Bytes, Physical Read Requests y CPU Time (ver figura 8).

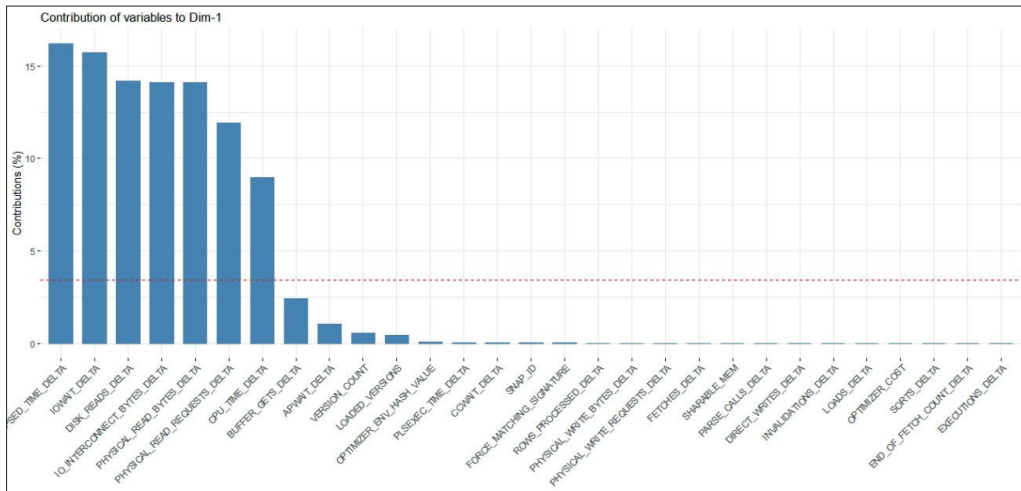


Figura 8. Variables representativas PCA1 de información de rendimiento de ejecuciones de sentencias SQL en base de datos DSS

Nota: "I/O Interconnect Bytes" fusiona valores de lectura y escritura en disco.

Elaboración propia

Para una carga de trabajo OLTP se identifican los siguientes factores en el componente principal 1 (PC1), donde se concentra la mayor parte de varianza del conjunto de datos: Elapsed Time, CPU Time, I/O Interconnect Bytes, Physical Read Bytes, Disk Reads, Physical Read Requests, IO Wait y Buffer Gets (ver figura 9).

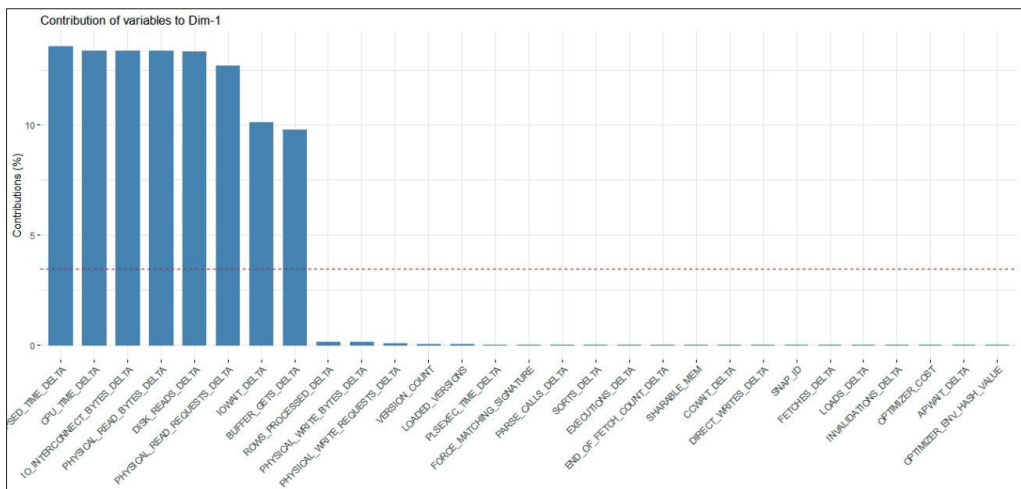


Figura 9. Variables representativas PCA1 de información de rendimiento de ejecuciones de sentencias SQL en base de datos OLTP

Elaboración propia

Para una carga de trabajo mixta (OLTP y DSS) se identifican los siguientes factores en el componente principal 1 (PC1), donde se concentra la mayor parte de varianza del conjunto de datos: Fetches, End of Fetch Count, Executions, Rows Processed, PLSExec time, CCWait, Elapsed Time y CPU Time (ver figura 10).

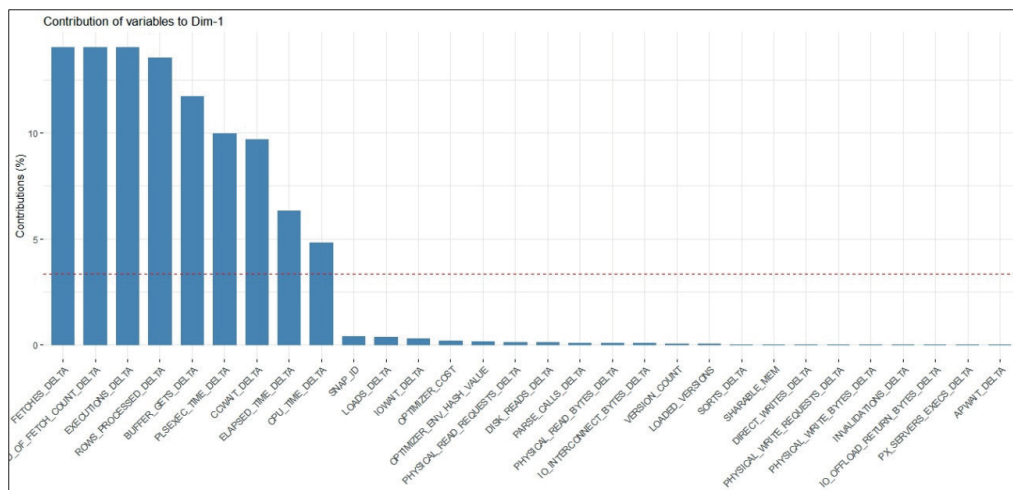


Figura 10. Variables representativas PCA1 de información de rendimiento de ejecuciones de sentencias SQL en base de datos con carga mixta OLTP y DSS
Elaboración propia

3.3 CCA entre las variables y factores resultantes de la reducción dimensional PCA

Aplicando el algoritmo basado en el análisis de correlación canónico, se establece que solo la primera dimensión canónica tiene nivel de significancia estadística con $p\text{-value} < 0,05$ (ver tabla 2), y los factores y variables que explican la mayor proporción de correlación entre ambos conjuntos son {Cost, CPU Cost, I/O Cost} y {IO Interconnect Bytes, Physical Read Bytes} (ver tabla 3).

Tabla 2
Dimensiones canónicas para la base de datos

| Tests of Canonical Dimensions - Canonical Correlation Analysis (CCA) | | | | | | |
|--|-----------|-------|-------|-----|----------|---------|
| DB Type | Dimension | Corr. | F | df1 | df2 | p-value |
| DSS | 1 | 0,87 | 14,35 | 84 | 3 907,96 | 0 |
| OLTP | 1 | 0,38 | 2,92 | 84 | 5 424,01 | 0 |
| OLTP y DSS | 1 | 0,35 | 2,09 | 84 | 7 218,74 | 0 |

Elaboración propia

Tabla 3
Correlación canónica de variables y factores por base de datos

| Tipo | Correlación | Plan de ejecución | Estadísticas y métricas de ejecución |
|------------|-------------|---------------------|--------------------------------------|
| DSS | 0,87 | Cost | IO Interconnect Bytes |
| | | CPU Cost | Physical Read Bytes |
| | | IO Cost | |
| OLTP | 0,38 | Bytes | Disk Reads |
| | | Cardinality | IO Interconnect Bytes |
| | | CPU Cost | Physical Read Bytes |
| OLTP y DSS | 0,35 | Cost | CPU Time |
| | | CPU Cost | Elapsed Time |
| | | Temp Space | End of Fetch Count |
| | | | Executions |
| | | | IO Interconnect Bytes |
| | | Physical Read Bytes | |

Elaboración propia

Por lo expuesto en la tabla 3, se llega al resumen de factores y variables influenciadas mostrado en la tabla 4.

Tabla 4

Factores que influyen en el rendimiento de las ejecuciones de sentencias SQL en sistemas RDBMS

| Factor | Estadística y métrica de ejecución correlacionada |
|---|--|
| <ul style="list-style-type: none"> • Cost • CPU Cost • IO Cost | <ul style="list-style-type: none"> • Physical Read Bytes (I/O Wait Time) • Physical Write Bytes (I/O Wait Time) • Disk Reads (I/O Wait Time) • CPU Time (Service Time) • Elapsed Time (Response Time) |

Elaboración propia

Se puede asociar cada factor encontrado con el componente asociado del tiempo de respuesta (Response Time), que es igual al tiempo de servicio (Service Time) más el tiempo de espera (Wait Time), tal como se muestra en la tabla 5.

Tabla 5

Factores que influyen en las categorías de rendimiento de las ejecuciones de sentencias SQL en sistemas RDBMS

| Factor que influye en el rendimiento | Categoría de rendimiento |
|--------------------------------------|-------------------------------------|
| Costo de CPU (CPU Cost) | Tiempo de Servicio (Service Time) |
| Costo de I/O (IO Cost) | Tiempo de Espera (Wait Time) |
| Costo del Plan (Cost) | Tiempo de Respuesta (Response Time) |

Elaboración propia

Asimismo, es posible cuantificar el impacto que tendrá en el tiempo de respuesta un perfil de costo determinado. Experimentalmente para bases de datos de un solo propósito, sea OLTP o DSS, si una sentencia SQL tiene un costo menor o igual a 822, entonces con 80 % de probabilidad se puede afirmar que su tiempo de respuesta al ejecutarse no excederá los 0,73 segundos. Esta información es útil al momento de establecer acuerdos de niveles de servicio.

4. CONCLUSIONES

Nuestro objetivo fue aportar a la investigación en este campo de estudio con una metodología que combina dos algoritmos de aprendizaje de máquina estadístico, *principal component analysis* y *canonical correlation analysis*, para conocer los factores que dominan o afectan la actuación o rendimiento de un sistema de gestión de base de datos relacional visto como caja negra, sin necesidad de entrar en los detalles internos de su arquitectura. Los resultados arrojan como factores influyentes al Costo, IO Cost y CPU Cost.

La efectividad de la metodología se apoya en contar con estimaciones de costo valederas y estas se obtienen cuando las estadísticas de los objetos del motor de base de datos están actualizadas, o en su defecto representan estadísticamente al conjunto de datos que se maneja. Esta aseveración es válida para cualquier implementación de base de datos relacional, como, por ejemplo, Oracle Database, IBM DB2 y Microsoft SQL Server. Se recomienda consultar respectivamente las siguientes referencias:

- i. https://docs.oracle.com/cd/B19306_01/server.102/b14211/stats.htm#i37048
- ii. https://www.ibm.com/support/knowledgecenter/es/SSEPEK_11.0.0/intro/src/tpc/db2z_queryandapplicationperformanceanalysis.html,
- iii. <https://docs.microsoft.com/es-es/sql/relational-databases/query-processing-architecture-guide?view=sql-server-ver15>

Verificar que el costo de una sentencia SQL, una magnitud proveniente de la ponderación de varios factores medidos en el momento de generar el plan de ejecución y donde predomina el componente de I/O en disco (IO Cost), es un factor que influye en el rendimiento de su ejecución. Puede inferirse que una sentencia SQL con costo alto (costo resultante en su plan de ejecución óptimo) tendrá generalmente un número considerable de operaciones en disco, lo que redundará en el aumento del tiempo de respuesta al momento de ejecutarse.

Los algoritmos que se usaron para el artículo son de la categoría de aprendizaje de máquina no supervisado. Se pueden incluir variantes kernelizadas de estos, a saber, kernel-PCA y kernel-CCA, para probar nuevas hipótesis y servir como plataforma de predicción de estadísticas y métricas de ejecución.

REFERENCIAS

- Badrinath Krishna, V., Weaver G.A., y Sanders W. H. (2015). PCA-Based Method for Detecting Integrity Attacks on Advanced Metering Infrastructure. En J. Campos y B. Haverkort (Eds.), *Quantitative Evaluation of Systems. 12th International Conference, QEST 2015, Proceedings* (pp. 70-85). Springer.
- Bontempi G., y Kruijtzter W. (2002). A Data Analysis Method for Software Performance Prediction. En *Proceedings of the Design, automation and test in Europe (DATE 02)* (pp. 971-976). IEEE.
- Brauckhoff, D., Salamatian, K., y May, M. (2009). Applying PCA for Traffic Anomaly Detection: Problems and Solutions. En *Proceedings of IEEE INFOCOMM 2009*.
- De, P., Sinha, A. P., y Vessey, I. (2001). An Empirical Investigation of Factors Influencing Object-Oriented Database Querying. *Information Technology and Management*, 2(1), 71-93.

- Fortier P. J., y Michel H. E. (2003). *Computer Systems Performance Evaluation and Prediction*. Digital Press.
- Ganapathi, A. S., Kuno, H. A., y Dayal, U. (2015). U. S. Patent No. 9,189,523. U.S. Patent and Trademark Office.
- Giusto P., Martin G., Harcourt E. (2001). Reliable Estimation of Execution Time of Embedded Software. *Proceedings of the DATE 2001 on Design, Automation and Test in Europe* (pp. 580-589). IEEE.
- Kleinrock, L. (1976). *Queuing Systems, Volume II: Computer Applications*. John Wiley & Sons.
- Mikolajczyk, K., Schmid, C., (2005). A performance evaluation of local descriptors. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 27(10), 1615-1630.
- Schkolnick, M., y Tiberio, P. (1985). Estimating the cost of updates in a relational database. *ACM Transactions on Database Systems (TODS)*, 10(2), 163-179.
- Shawe-Taylor, J. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.
- Smith, C. U., y Williams, L. G. (2002). Performance and Scalability of Distributed Software Architectures: An SPE Approach. *Scalable Computing: Practice and Experience*, 3(4).
- Zhang L., Liu F., y Zeng Z. (2018). Combining Link and Content Correlation Learning for Cross-Modal Retrieval in Social Multimedia. En Q. Zu y B. Hu (Eds.), *Human Centered Computing, HCC 2017. Lecture Notes in Computer Science*, 10745. Springer, Cham.