
16

ANÁLISIS DE RENDIMIENTO DE LAS TECNOLOGÍAS PLINQ Y LINQ EN SISTEMAS INFORMÁTICOS.

*José Guerra Salazar, Julio Santillán Castillo,
José Silva Monard*

*Fecha de recepción: 26 de agosto del 2015
Fecha de aceptación: 18 de noviembre del 2015*

ANÁLISIS DE RENDIMIENTO DE LAS TECNOLOGÍAS PLINQ Y LINQ EN SISTEMAS INFORMÁTICOS

José Guerra Salazar¹, Julio Santillán Castillo¹, José Silva Monard¹

¹Escuela Superior Politécnica de Chimborazo (ESPOCH)
Panamericana Sur kilómetro 1 ½
Riobamba- Ecuador
E-mail: j_guerra@epoch.edu.ec

Resumen

La investigación se basó en analizar el rendimiento de los lenguajes integrados de consulta de datos LINQ y PLINQ, para el desarrollo de sistemas informáticos. La medición del rendimiento contempla cuatro indicadores: Tiempo de respuesta, porcentaje de uso de procesamiento, uso de memoria RAM, y operaciones de entrada y salida a discos duros. Los valores correspondientes a los indicadores son obtenidos a través de un módulo software incrustado en el prototipo desarrollado, mismo que a la vez integra LINQ y PLINQ. Con los datos de cada indicador y aplicando estadística descriptiva e inferencial, se determinó que PLINQ ofrece un mejor rendimiento (81.75%) que el lenguaje integrado de consultas LINQ (61.50%). En la ejecución de consultas complejas se obtuvo una diferencia de 61.25%, mientras que LINQ en consultas de datos simples, resalta su superioridad con 33%.

Palabras Claves: Análisis comparativo, lenguajes integrados de consulta, rendimiento, LINQ, PLINQ.

Abstract

The research is based on analyzing the performance of integrated data query languages LINQ and PLINQ for computer systems development. The performance measurement includes four indicators: response times, percentage of use of processing, RAM usage, and input and output operations to hard drives. The values for the indicators are obtained through a software module embedded in the prototype developed the same that integrates both LINQ and PLINQ. With data for each indicator and applying descriptive and inferential statistics, it was determined that PLINQ offers better performance (81.75%) than the integrated LINQ queries (61.50%) language. In the execution of complex queries a difference of 61.25% was obtained, while simple LINQ query data, highlighting its superiority with 33%.

Keywords: Comparative analysis, integrated query languages, performance, LINQ, PLINQ.

1. Introducción

En aplicaciones software existen diversas formas de gestionar datos dentro de la aplicación, y entre estas los lenguajes integrados de consultas, y estos agregan capacidades de consulta y manipulación de datos a los lenguajes de programación de propósito general, como LINQ (Language Integrated Query por sus siglas en inglés) y PLINQ (Parallel Language Integrated Query, por sus siglas en inglés). Por su parte PLINQ combina la simplicidad y legibilidad de la sintaxis de LINQ con la eficacia de la programación paralela [6], aprovechando los recursos hardware del equipo del usuario final.

El bajo rendimiento de las aplicaciones software comúnmente se presenta en el procesamiento de grandes cantidades de datos, puesto que la manipulación de dichos datos está realizada con lenguajes de consulta que no brindan un rendimiento adecuado, por lo que LINQ y PLINQ ofrecen grandes oportunidades en cuanto a la consulta y manipulación de datos, pero su uso escenarios desfavorables disminuye rendimiento general en el producto final. Aún cuando existen recomendaciones sobre el uso de estos lenguajes en determinados casos, se desconocen estudios científicos que sugiera la utilización específica de LINQ o PLINQ.

Esta investigación busca determinar cuál de los dos lenguajes integrados de consultas PLINQ o LINQ presenta un mejor rendimiento en cuanto a velocidad y a uso de recursos hardware.

2. Materiales y métodos

Para el desarrollo de esta investigación se implementó un sistema de escritorio con la finalidad de estudiar las características y prestaciones de los lenguajes integrados de consultas PLINQ y LINQ, establecer los parámetros, herramientas e indicadores para medir el rendimiento de las aplicaciones de Software

2.1 Definiciones

Un lenguaje integrado de consultas es una extensión a un lenguaje de programación, que permite la consulta y manipulación de los datos que procesa el software. El mayor exponente de los lenguajes integrados de consulta es LINQ, junto con su implementación en paralelo PLINQ.

LINQ es un lenguaje integrado de consultas para la construcción de aplicaciones dirigidas por datos que permite su extracción desde una fuente hacia una aplicación [2]. Busca la eficacia de la extracción y

tiempos de respuesta bajos en comparación de las técnicas tradicionales.

PLINQ es un motor de consultas LINQ que hace uso de varios procesadores o núcleos en paralelo [4] para su ejecución. Se emplea en consultas de sólo lectura que procesan grandes cantidades de datos [1].

2.2. Características de LINQ y PLINQ

En la Tabla I se muestra las características por cada uno de los lenguajes integrados de consultas. De modo que se puede evidenciar la diferencia entre dichos lenguajes.

Tabla I. Características de LINQ y PLINQ

Características	LINQ	PLINQ
Integración con XML	Si	Si
Integración con SQL	Si	Si
Integración con DataSets	Si	Si
Integración con Objetos	Si	Si
Integración con Entity Framework	Si	Si
Paralelización de Consultas	No	Si
Manipulación de Datos, desde Bases de datos.	No	Si

Fuente: Grupo de investigación, basado en información procesada

La característica de manipulación de datos desde bases de datos no está presente en LINQ, debido a que este no soporta métodos para manipulación de datos, porque no se pueden transformar en arboles de expresión, así como también en consultas T-SQL [5].

2.3. Análisis de rendimiento

El rendimiento del software se enfoca en agrupar indicadores con los cuales se puede emitir juicios objetivos acerca de su funcionalidad [5]. El rendimiento se refiere principalmente al *tiempo* de respuesta de la aplicación [8], y puede involucrar indicadores como: el grado de uso de los recursos, la cantidad de llamadas a funciones de sistema, o el esfuerzo en la ejecución de las tareas. Es decir,

dependiendo del contexto, el rendimiento podrá acarrear diversas métricas con un fin común.

2.4. Parámetros

Dada la naturaleza de esta investigación y las diferencias encontradas entre los lenguajes integrados de consulta LINQ y PLINQ, se han definido dos parámetros: Fuente Externa y Fuente Interna. Dichos parámetros se describen como:

Fuente Externa (Fe): consiste en medir el rendimiento cuando la aplicación o prototipo emplee datos que no residan junto a la aplicación (prototipo). Para la investigación, su ponderación es de 30%, puesto que, al requerir software de terceros para extraer datos (librerías de lectura de archivos, ODBC, OLE, entre otros), no controla el nivel de “paralelismo” que se maneja dentro de dichos software [7], y de esta manera se pueden generar “cuellos de botella”.

Fuente Interna (Fi): consiste en medir el rendimiento cuando el prototipo usa una fuente de datos residente en memoria junto a la aplicación. Para la investigación su ponderación es del 70% debido a que este parámetro no incurre en el uso de software de terceros.

Para la Fuente Interna y Externa se contemplaron sub-parámetros de consultas simples con una ponderación del 40%, y consultas complejas con ponderación del 60%.

2.5. Indicadores

Para la investigación se consideran los siguientes indicadores para medir el rendimiento:

- Tiempo de respuesta del software.- Es el tiempo transcurrido desde la solicitud hasta la entrega del resultado. Su ponderación se establece con el 75%, debido a que se considera una métrica imprescindible de rendimiento [3].
- Porcentaje de uso de procesamiento.- Es el porcentaje de uso de núcleos de procesamiento, durante una petición al sistema. Se establece una ponderación del 10%.
- Cantidad de memoria RAM empleada.- Cantidad promedio de memoria RAM utilizada por la aplicación durante el procesamiento de una petición. Tiene una ponderación del 10%.
- Operaciones de Lectura/Escritura en Discos.- Cantidad promedio de operaciones I/O a discos duros. Se define una ponderación del 5%.

2.6. Modelo matemático del rendimiento.

El rendimiento se puede expresar a través de un modelo matemático, para el cual, en esta

investigación, se le han incrustado variables que identifican a los parámetros e indicadores antes descritos.

Tal que la magnitud que identifique al rendimiento (R) se expresa por

$$R = 0.7 * F_{interna} + 0.3 F_{externa}$$

En donde se entiende que el 70% del rendimiento corresponde al cálculo con datos desde fuente interna, y el restante 30% a fuente externa.

La magnitud que corresponde a ‘Fuente’ (Fuente de datos externas o internas), se expresa como la suma de las magnitudes correspondientes a consultas simples (Cs) y consultas complejas (Cc), definidas en la siguiente ecuación:

$$Fuente = 0.4 * Cs + 0.6 Cc$$

Para determinar el valor de las ‘Consultas’ (C) se considera que es igual a la suma de los indicadores ‘Tiempo’ (T), ‘CPU’, ‘RAM’ y ‘Disco’, definiéndose en ecuación como:

$$C = 0.75T + 0.1CPU + 0.1RAM + 0.05 Disco$$

2.7. Herramientas

La herramienta utilizada para la mediación del rendimiento es un módulo de software incorporado en el prototipo de pruebas. Utiliza el API de Windows correspondiente a ‘Performance Monitor’. La Figura 1 muestra el funcionamiento de la medición del rendimiento dentro del prototipo de software.

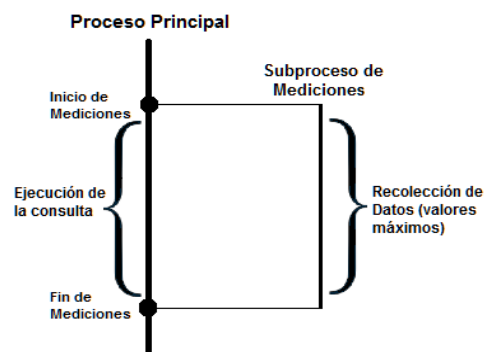


Figura 1. Medición del rendimiento
Fuente: Grupo de investigación, basado en información procesada

2.8. Descripción del Experimento.

Con la ayuda del prototipo (Software) implementado en LINQ y PLINQ, se ejecutarán procesos de consultas y manipulaciones de datos. Se toman lecturas de los indicadores de rendimiento mientras se realizan estos procesos. Se infiere una calificación en porcentaje, aplicando una regla de tres inversa al

valor de la media (\bar{X}) de cada indicador. El análisis estadístico para determinar si la variación de los datos es significativa. Con los porcentajes finales es posible calcular la magnitud que identifique al rendimiento de acuerdo a las fórmulas planteadas.

3. Resultados

Dados los valores correspondientes a cada indicador se calcula el valor con las formulas detalladas en los acápite anteriores. La Tabla II corresponde a la recopilación y análisis de datos.

Tabla II. Calificaciones obtenidas

		Fuente Interna		Fuente Externa	
		Consulta Simple	Consulta Compleja	Consulta Simple	Consulta Compleja
LINQ	Tiempo	100%	37.1%	100%	0%
	CPU	100%	100%	100%	0%
	RAM	100%	100%	99.2%	0%
	Disco	100%	100%	51.4%	0%
PLINQ	Tiempo	72.6%	100%	57.3%	100%
	CPU	45.1%	13.7%	18.4%	100%
	RAM	73.6%	46.7%	100%	100%
	Disco	100%	100%	100%	100%

Fuente: Grupo de investigación, basado en información procesada

De modo que como resultado de la aplicación del modelo matemático se tiene los valores mostrados en la Tabla III.

Tabla III. Resultados del análisis de rendimiento

		LINQ	PLINQ
Fuente Interna	Consulta Simple	100%	71.30%
	Consulta Compleja	52.87%	86.03%
	Total Fuente	71.72%	80.14%
Fuente Externa	Consulta Simple	97.49%	59.87%
	Consulta Compleja	0%	100%
	Total Fuente	38.99%	83.94%
Total Lenguaje		61.90%	81.28%

Fuente: Grupo de investigación, basado en información procesada

En la Figura 2 se puede apreciar los resultados generales de forma gráfica (resultados con respecto a fuentes de datos).

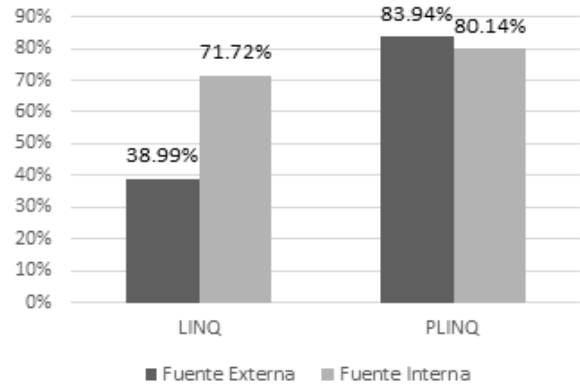


Figura 2. Resultados categorizados por fuente de datos

Fuente: Grupo de investigación, basado en información procesada

Además, los datos consolidados con respecto a tipos de consulta se muestran en la Tabla IV y en el gráfico de la Figura 3. Con dichos datos (promedio entre fuente interna y fuente externa) se puede evidenciar el entorno favorable para cada lenguaje integrado de consultas.

Tabla IV. Resultados por tipo de consulta

	Consulta Simple	Consulta Compleja
LINQ	98.75%	26.44%
PLINQ	65.57%	93.02%

Fuente: Grupo de investigación, basado en información procesada

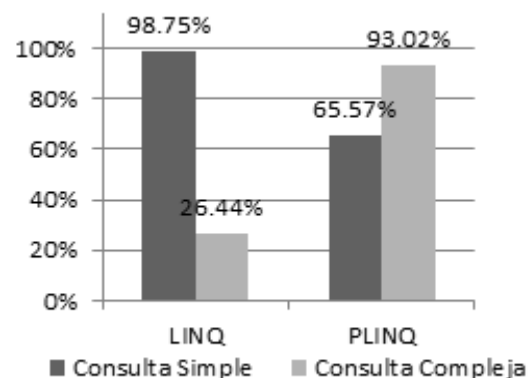


Figura 3. Resultados categorizados por tipo de consulta.

Fuente: Grupo de investigación, basado en información procesada

Mientras que en la Figura 4 se muestra los resultados totales en cuanto a rendimiento, y se puede evidenciar que PLINQ posee la mayor magnitud en cuanto a rendimiento (19.38 puntos más que LINQ), de modo que se acepta la hipótesis de que “La utilización del lenguaje integrado de consultas PLINQ ofrece un mejor rendimiento en la manipulación de datos en una aplicación informática; en relación al lenguaje integrado de consultas LINQ, en entornos de gran capacidad de procesamiento”. (Los resultados están sujetos a un error del 5%, dado que la significación estadística en el análisis fue del 95%).

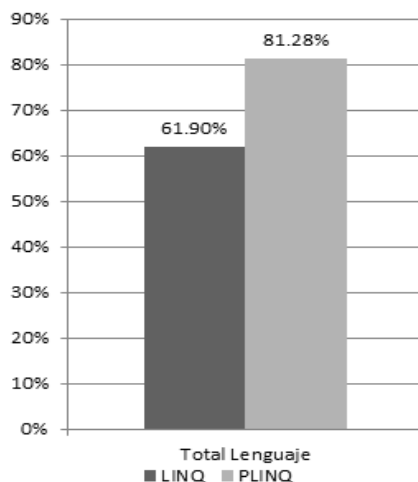


Figura 4. Rendimiento en los lenguajes integrados de consulta

Fuente: Grupo de investigación, basado en información procesada

4. Conclusiones

- El lenguaje integrado de consultas en paralelo PLINQ ofrece un mejor rendimiento (19.38% mejor) que LINQ, en entornos donde existan transacciones complejas y de gran volumen de datos.
- Cuando se trata de consultas de datos en donde no se apliquen transformaciones ni filtros demasiado complejos, LINQ ofrece un rendimiento promedio de 33.18% más que PLINQ.
- En consultas complejas se determina que el rendimiento promedio de PLINQ es 66.58% mejor que LINQ.
- Cuando se trabaja sobre fuentes de datos internas PLINQ ofrece un mejor rendimiento (un 8.42% mejor que LINQ), al igual que en fuentes externas (un 44.95% mejor que LINQ).

- Al analizar consultas simples de fuente externa, en uso de disco, PLINQ supera en 48.60 % a LINQ, y con el mismo porcentaje, en el indicador Tiempo de respuesta de software, ocurre lo contrario.
- En la operación de lectura y escritura en disco, PLINQ obtiene la máxima valoración, independientemente del tipo de consulta y fuente de datos. LINQ presenta un comportamiento similar, a excepción de las consultas simples de fuente externa, en la que presenta el valor más bajo, equivalente a 51.40%.
- LINQ, en consultas simples y complejas, con datos de fuente interna, presenta la máxima valoración porcentual en cuanto a tiempo de respuesta, uso de procesamiento y cantidad de memoria RAM empleada.
- En el indicador porcentaje de uso de procesamiento, la valoración porcentual más baja, 13.70%, la obtiene PLINQ en consultas complejas con datos de fuente interna, coincidentemente con el indicador cantidad de memoria RAM empleada, en que se obtiene 46.70%.

5. Bibliografía

- [1] **Alba, Wilson & Sánchez, Adilén. 2010.** PLINQ. Motor de ejecución de consultas paralelas. Serie Científica de la Universidad de las Ciencias Informáticas, Vol. 3, No. 12, 2010, Cuba. 14p.
- [2] **Barragán, Rodrigo & Paguay, Paul. 2010.** Estudio del Lenguaje Integrado de Consultas (LINQ), Aplicado al Desarrollo del sistema de Inventario y Facturación de la Librería Politécnica. (Tesis). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería en Sistemas.
- [3] **Castillo, Jessica. 2012.** Estudio Comparativo del Rendimiento de Servidores Web de Virtualización Sobre la Plataforma Windows Server 2008. (Tesis). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes. Riobamba-Ecuador. 2012, pp. 79-100; 130-152.
- [4] **Duffy, Joe., & Essey, Ed. 2007.** Ejecución de consultas en procesadores multinúcleo. MSDN

Magazine. Vol. 1, No. 92, Octubre 2007, US.
pp. 17-26.

- [5] **Microsoft. 2006.** El proyecto LINQ. Tomado de <https://msdn.microsoft.com/es-es/library/bb308959.aspx>.

- [6] **Microsoft. 2014.** Parallel LINQ (PLINQ). MSDN Developer Network. Tomado de <https://msdn.microsoft.com/es-us/library/dd460688%28v=vs.110%29.aspx>.

- [7] **Toub, Stephen, & Shafi, Hazim. 2008.** Mayor compatibilidad con el paralelismo en la próxima versión de Visual Studio. MSDN Magazine. Vol. 1, No. 104, Octubre 2008, US. pp. 9-12.

- [8] **Weidendorfer, Josef & Zenith, Federico. 2015.** Manual de KCachegrind. Tomado de <https://docs.kde.org/trunk4/eskdesdk/kcachegrind/index.html>.