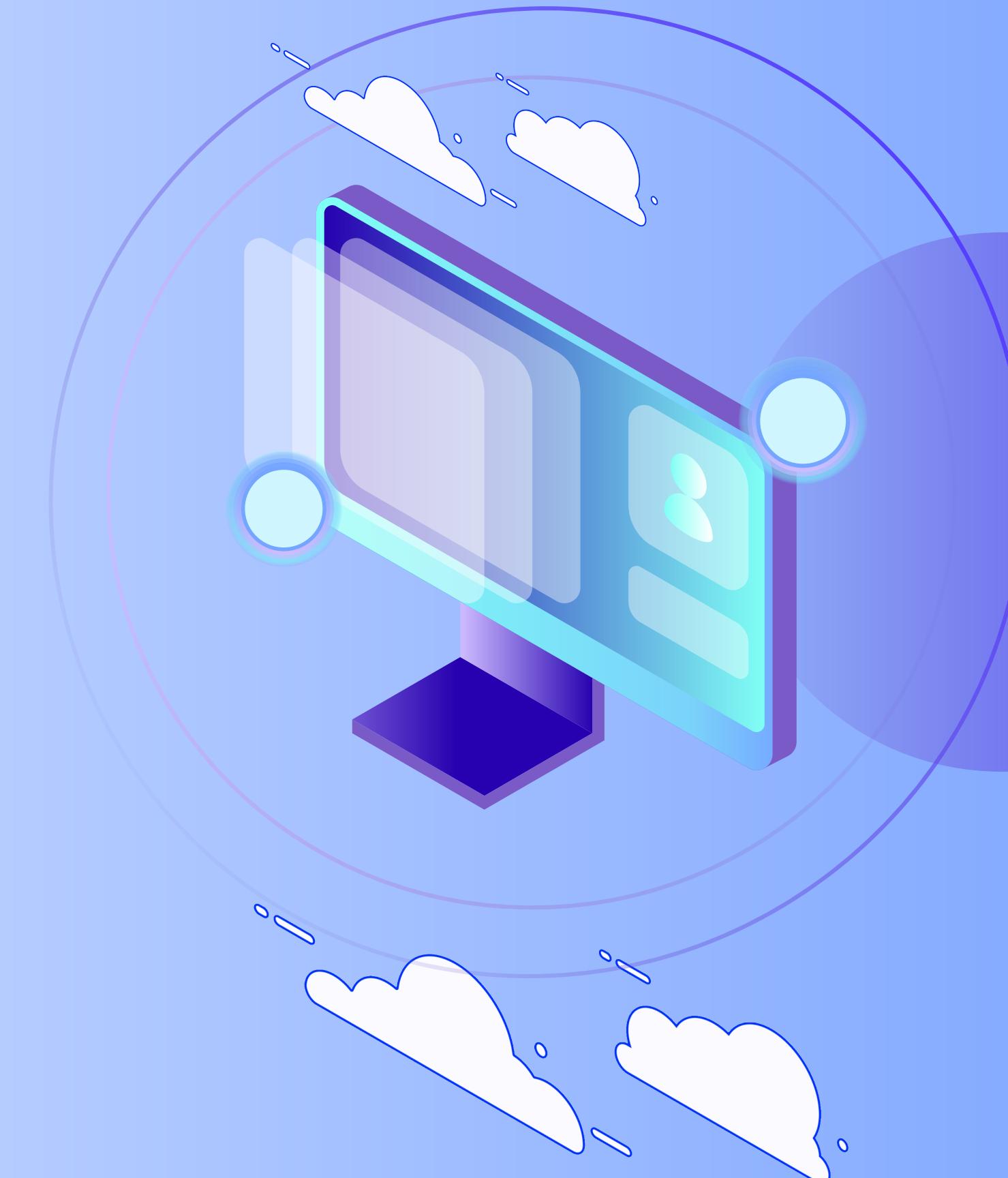


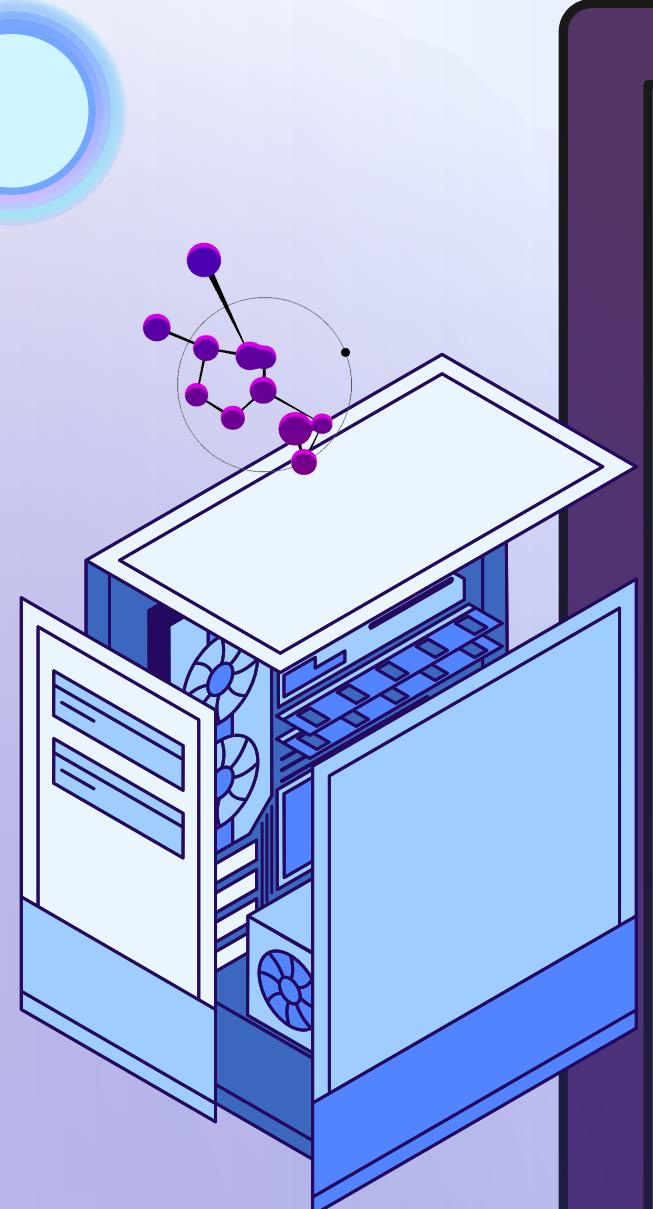
COMPARACIÓN DE TÉCNICAS DE COMPUTACIÓN PARALELA, CONTENEDORES Y CLOUD COMPUTING

PARA ACELERAR PROCESOS DE WEB SCRAPING

Integrantes: Montalvo Fabrizio, Salazar Paolo, Zhou Cynthia



INTRODUCCIÓN



El volumen de datos digitales superará los 180 zettabytes para 2025, con millones de sitios web actualizándose diariamente (Domo, 2023). El web scraping es esencial, pero las estrategias secuenciales tienen limitaciones.

Enfoques modernos

- Computación paralela: Distribuye tareas para reducir tiempos.
- Contenedores (Docker): Entornos portátiles y eficientes.
- Computación en la nube: Recursos escalables bajo demanda.

OBJETIVO

Aplicar las técnicas mencionadas en la extracción de datos de fútbol, evaluando tiempo de ejecución, uso de CPU/memoria, escalabilidad y facilidad de implementación.

Evaluar y comparar computación paralela, contenedores y computación en la nube para optimizar procesos de web scraping, usando como caso de estudio la extracción de datos de jugadores de fútbol desde una web pública.

ESTADO DEL ARTE

On Multi-Thread Crawler Optimization for Scalable Text Searching

Guang Sun, Huanxin Xiang y Shuanghu Li (2019)



Optimizador Multihilo para Crawlers

Proponen un optimizador multihilo para crawlers, combinando técnicas de búsqueda en anchura (BFS) y en profundidad (DFS). Sus experimentos en Wikipedia demostraron una reducción drástica del tiempo de scraping, de 1000 a 273 segundos, validando la eficiencia del multithreading.



ESTADO DEL ARTE

A Cloud-based System for Scraping Data from Amazon Product Reviews at Scale

Ryan Woodall, Douglas Kline, Ron Vetter y Minoo Modaresnezhad (2022)



Sistema de Scraping Basado en la Nube

Presentan una solución en la nube para extraer reseñas de Amazon. Utilizan un servicio de scraping de terceros, Azure Functions para comunicación y Azure Data Lake para almacenamiento. Su arquitectura de microservicios destaca por su modularidad, escalabilidad y facilidad de integración con flujos ETL.



ESTADO DEL ARTE

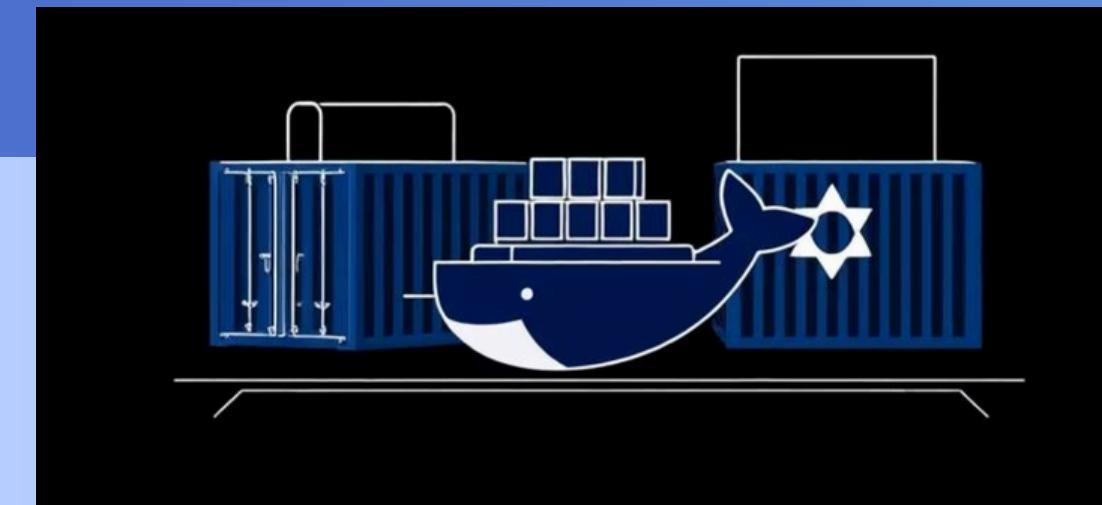
A Cloud-based System for Scraping Data from Amazon Product Reviews at Scale

Ryan Woodall, Douglas Kline, Ron Vetter y Minoo Modaresnezhad (2022)



Escalabilidad Horizontal con Contenedores

Exploran una arquitectura de web crawler escalable horizontalmente utilizando Docker y Kubernetes. Cada contenedor ejecuta una instancia aislada del scraper, coordinada desde una interfaz gráfica que permite definir parámetros como la profundidad del crawling y palabras clave.





MARCO TEÓRICO

Web Scraping

Técnica de extracción automatizada de datos desde sitios web. Utiliza programas que navegan por páginas web y recogen información útil como precios, descripciones o contenido específico.

MARCO TEÓRICO



Parallel computing

Modelo de procesamiento donde se realizan múltiples operaciones simultáneamente usando varios procesadores o núcleos del dispositivo. Optimiza el tiempo de ejecución de las tareas al distribuir la carga de trabajo.

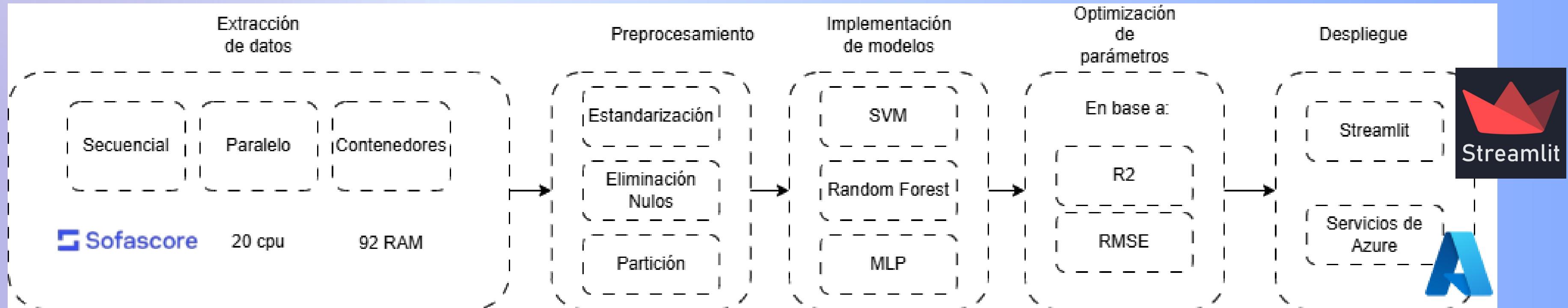
Docker containers

Tecnología que permite ejecutar aplicaciones en entornos aislados con todas sus dependencias, sin afectar el sistema operativo . Docker es la herramienta más popular para implementar contenedores.

Cloud computing

Modelo de entrega de servicios que ofrece recursos informáticos a través de Internet. Los usuarios pueden acceder a estos servicios desde cualquier lugar, sin necesidad de tener una infraestructura costosa.

METODOLOGÍA

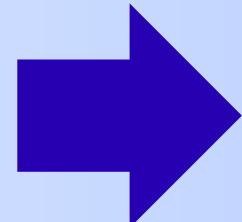


METODOLOGÍA



Scraping Secuencial

- Script en Python como línea base
- Librerías: BeautifulSoup, Selenium
- No optimización, sin paralelismo

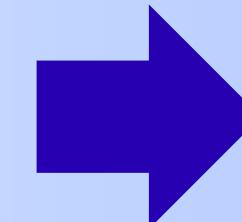
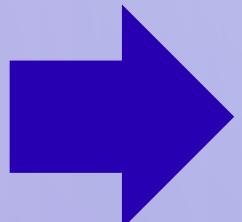


Scraping Paralelo Local

- Uso de ThreadPoolExecutor y ProcessPoolExecutor
- Configuraciones: 10, 15 y 20 workers
- Evaluación de impacto del paralelismo

Contenerización con Docker

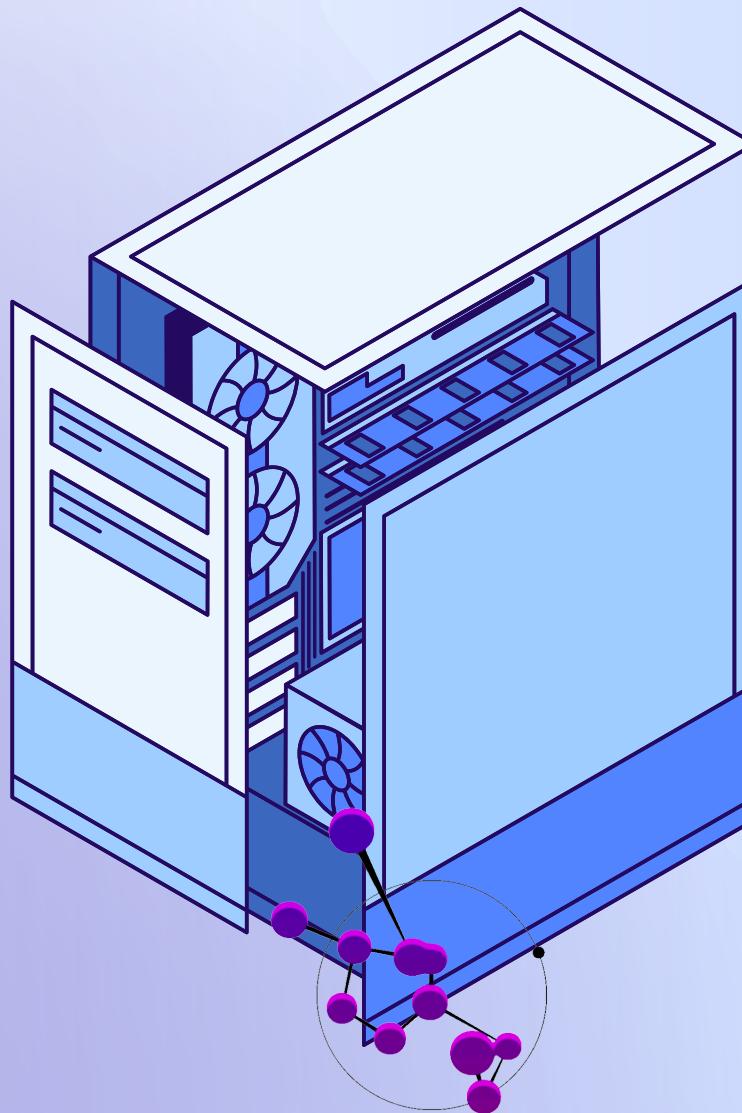
- Imagen Docker con dependencias
- Ejecución simultánea de múltiples contenedores
- Escenarios: 1, 3, 5 y 10 contenedores
- Evaluación de escalabilidad horizontal



Despliegue en la Nube (Azure)

- Modelo de Machine Learning
- Streamlit

ESPECIFICACIONES



- **Procesador:** Intel Core i9-7900X
- **Sistema Operativo:** Windows 11
- **Memoria RAM:** 96 GB
- **Cantidad de procesadores:** 20

EXPERIMENTOS Y RESULTADOS

	Scraping secuencial	Intento 1 Scraping paralelo	Intento 2 Scraping paralelo	Intento 3 Scraping paralelo
Workers	1	20	15	10
Tiempo	4:40 horas	No empezó	No terminó	1:20 horas
CPU promedio	9.1%	?	100%	16.4%
RAM promedio	26.4%	?	?	23.1%

$$\text{Speedup} = \frac{4.67 \text{ h}}{1.33 \text{ h}} \approx 3.51$$

CONCLUSIONES

El uso de técnicas de paralelismo local, como los hilos y procesos múltiples, demostró una mejora significativa en el rendimiento del scraping. Al reducir el tiempo de ejecución de 280 minutos a solo 80, se evidencia que dividir la carga de trabajo entre varios núcleos permite optimizar la eficiencia sin requerir hardware adicional.

CONCLUSIONES

El uso de múltiples entornos de ejecución independientes demostró ser una estrategia efectiva para escalar horizontalmente el scraping. Al dividir el conjunto de datos en segmentos asignados a distintas instancias, se logró una mejor distribución de la carga de trabajo, aumentando la eficiencia y facilitando el control del procesamiento paralelo.

CONCLUSIONES

Las pruebas realizadas en diferentes entornos permitieron validar que las mejoras en rendimiento no dependen únicamente del código, sino también de cómo se estructura y despliega la solución. La elección de una técnica adecuada debe considerar factores como la cantidad de datos a procesar, la disponibilidad de recursos y la necesidad de escalar el sistema de forma flexible y controlada.

REFERENCIAS

- [1] Domo Inc., Data Never Sleeps 11.0, 14 de diciembre de 2023.
- [2] R. Mitchell, Web Scraping with Python: Collecting More Data from the Modern Web, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
- [3] A. Grama, A. Gupta, G. Karypis, y V. Kumar, Introduction to Parallel Computing, 2^a ed., Boston, MA, EE. UU.: Addison-Wesley, 2003.
- [4] D. Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment," *Linux J.*, vol. 2014, no. 239, Art. no. 2, Marzo 2014. [En línea]. Disponible en: <https://dl.acm.org/doi/10.5555/2600239.2600241>
- [5] M. Armbrust et al., "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Abril 2010. [En línea]. Disponible en: <https://doi.org/10.1145/1721654.1721672>
- [6] G. Sun, H. Xiang, y S. Li, "On Multi-Thread Crawler Optimization for Scalable Text Searching," *Journal on Big Data*, vol. 1, no. 2, pp. 89–106, 2019. [En línea]. Disponible en: <https://pdfs.semanticscholar.org/1545/432272c6c4c352416f7ad29bcfb26550d840.pdf>
- [7] R. Woodall, D. Kline, R. Vetter, y M. Modaresnezhad, "A Cloud-based System for Scraping Data from Amazon Product Reviews at Scale," *Journal of Information Systems Applied Research*, vol. 15, no. 3, pp. 24–31, 2022. [En línea]. Disponible en: <https://jisara.org/2022-15/n3/JISARv15n3.pdf#page=24>
- [8] A. Prusty, O. Mejia, A. Shah, P. Kancherlapalli, A. Suresh, y R. Schiebel, "Horizontally Scalable Web Crawler using Containerization and a Graphical User Interface," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 5, pp. 292–297, 2020. [En línea]. Disponible en: <https://pdfs.semanticscholar.org/b8dd/7f8a92065fbb93b1306b37730096303067fa.pdf>
- [9] M. A. Khder, "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application," *Int. J. Advance Soft Compu. Appl.*, vol. 13, no. 3, pp. 145–160, 2021. [En línea]. Disponible en: doi.org/10.15849/IJASCA.211128.11
- [10] W. P. Petersen y P. Arbenz, *Introduction to Parallel Computing*, Oxford University Press, 2004.
- [11] E. Casalicchio and V. Perciballi, "Measuring Docker Performance: What a Mess!!!," in Proceedings of the ICPE '17 Companion, L'Aquila, Italy, 2017, pp. 81–84, [En línea]. Disponible en: doi.org/10.1145/3053600.3053605.
- [12] S. P. Mirashe and N. V. Kalyankar, "Cloud Computing," *Journal of Computing*, vol. 2, no. 3, pp. 78–82, Mar. 2010. [En línea]. Disponible en: <https://arxiv.org/abs/1003.4074>

The background features a large, semi-transparent purple circle on the left and a smaller, solid blue circle on the right, both with thin white outlines. A small, glowing blue circular element is positioned at the bottom center.

**¡MUCHAS
GRACIAS!**