

# ANÁLISIS DE CLAVES DE FRAGMENTACIÓN BASADAS EN RANGO PARA MONGODB

Bases de Datos no Relacionales 2021

Juan Miranda / Martín Calcagno

# Fragmentación y claves de fragmento

La fragmentación consiste en dividir una colección, distribuyendo sus datos en varios servidores.

Para fragmentar una colección, se debe elegir una clave de fragmentación.

Existen dos tipos de clave de fragmentación:

- **Clave de hash:** se ejecuta una función de hash sobre el valor de la clave, devolviendo el shard al que pertenece dicho valor.
- **Clave de rango:** los valores de la clave son divididos en rangos, de forma de que a cada rango que corresponda un fragmento.

# Elegir una clave de fragmento adecuada

Según la documentación de MongoDB, una buena clave debe tener 3 propiedades:

- **Uniformidad:** ayuda a que los shards se vayan 'llenando' de forma pareja.
- **Aleatoriedad:** para que las operaciones de escritura se propaguen a shards distintos, evitando sobrecargas de uno o pocos shards.
- **Localidad:** para que las operaciones de lectura puedan ser resueltas accediendo a pocos shards, evitando sobrecargas.

# Bases del trabajo realizado

Este trabajo fue basado en el Paper:

*"Analysis of range-based key properties for sharded cluster of MongoDB"*  
realizado por Kookarinrat y Temtanapat (Tailandia).

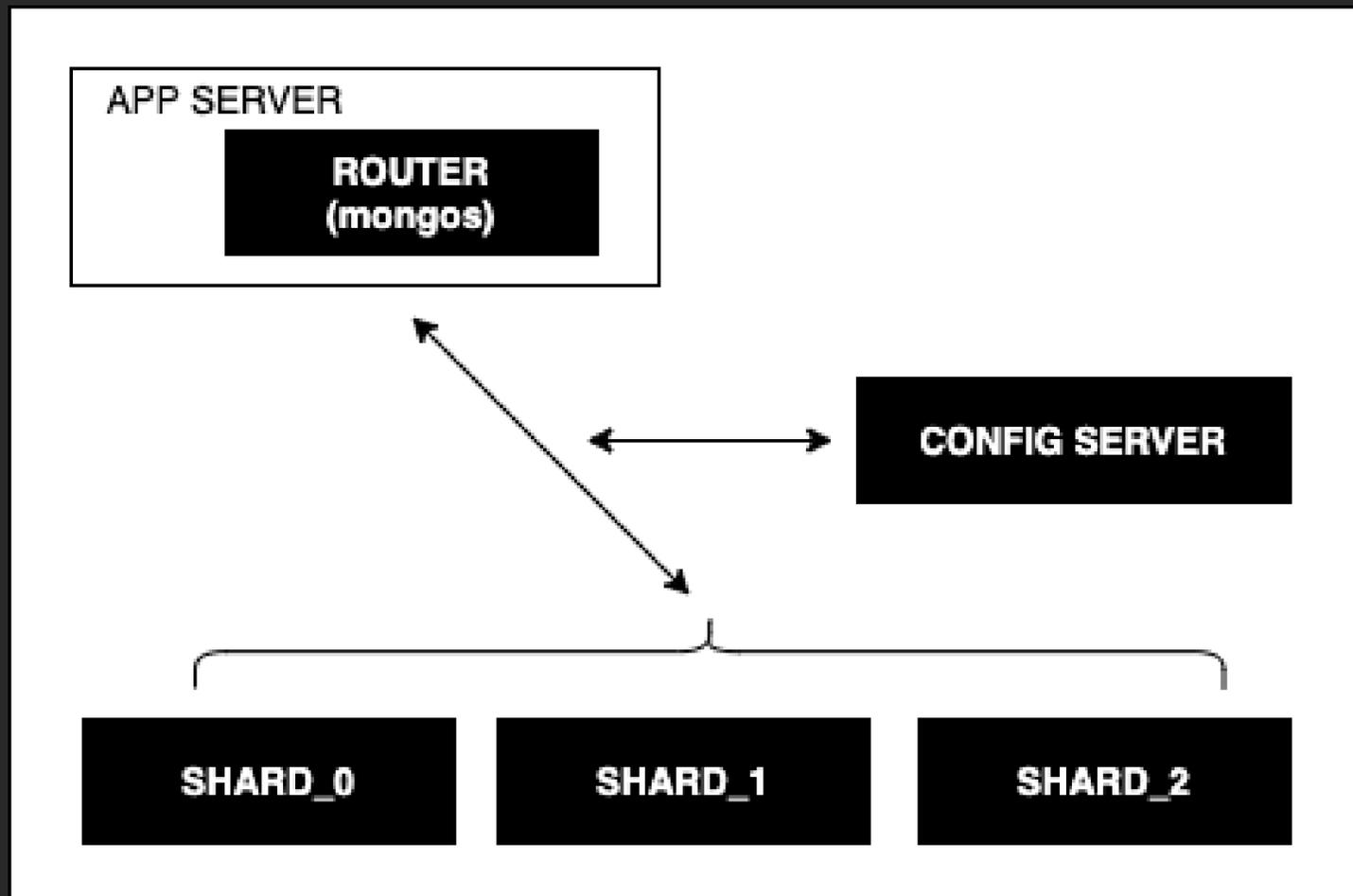
Su objetivo es analizar y evaluar las distintas propiedades de las claves de fragmentación sugeridas por MongoDB.

Basan sus estudios en claves de fragmentación con partición por rango.

# Objetivos de este trabajo:

- Profundizar en el uso y configuraciones de MongoDB.
- Aprender sobre fragmentación en MongoDB.
- Utilizar Docker.
- Replicar los experimentos realizados en el trabajo mencionado y comparar los resultados obtenidos.

# Infraestructura creada



- **Servidor de configuración:** Este componente de configuración recibirá solicitudes de configuración de lectura y escritura de un enrutador.
- **Servidor de enrutamiento:** Un enrutador es un punto de conexión a los clusters de fragmentos. Es el encargado de recibir las consultas, enviarlas a los diferentes fragmentos, recibir sus respuestas, y combinarlas para retornar los datos al cliente.
- **Servidor de fragmentos:** Un fragmento es un componente que almacena parte de los datos.

Los ambientes de trabajo se levantaron en Docker, con una máquina virtual por servidor.

# Configuración de claves de fragmentos

Esquema clave de fragmentos	
Clave	Fragmento
$0 \leq \text{clave} < 333.333$	Shard 0
$333.333 \leq \text{clave} < 666.666$	Shard 1
$666.666 \leq \text{clave} < 1.000.000$	Shard 2

# Conjunto de datos

Se obtuvieron de la API de Twitter una cantidad de datos significativa, contando con: tweets, retweets, y notificaciones de borrado de tweets.

Estos datos fueron filtrados dejando solamente 1 millón de documentos de tipo 'tweet', con los datos más importantes:

- El texto.
- Los hashtags utilizados.
- La fecha de publicación.
- Datos del usuario: nombre, cantidad de seguidores, cantidad de seguidos.

# Experimentos de escritura

Se realizaron pruebas de escritura con distintas configuraciones de claves:

- **Random shard key:** clave aleatoria.
- **Sorted shard key:** clave ordenada.
- **Combined key:** combinación de ambas.

# Tipos de experimentos para escritura

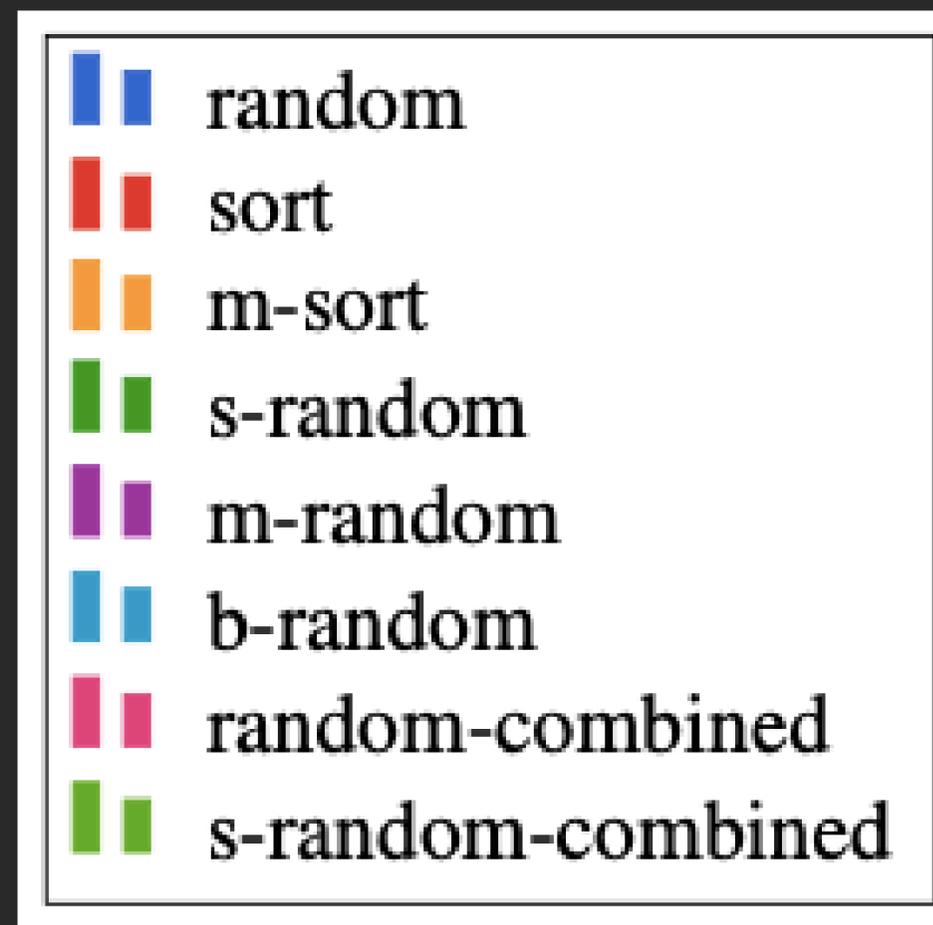
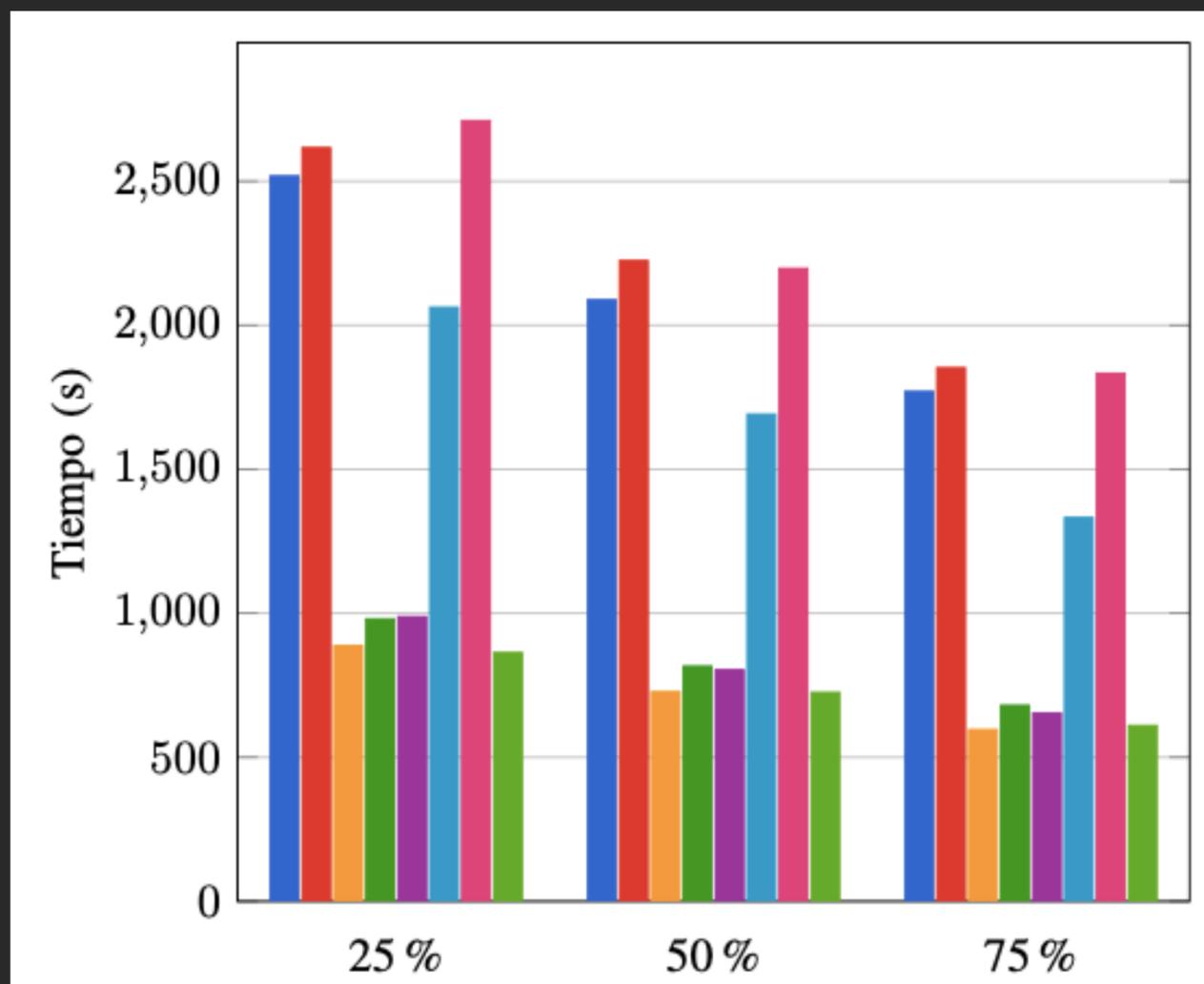
<b>Esquema de las colecciones para las pruebas</b>			
<b>Nombre</b>	<b>Ordenado</b>	<b>Aleatorio</b>	<b>Cantidad</b>
sort	X		1,000,000
m_sort	X		333,333
random		X	1,000,000
s_random		X	333,333
m_random		X	333,333
b_random		X	666,666
random_combined	X	X	1,000,000

# Ejecución de experimentos de escritura

Por cada uno de los experimentos anteriores se crea un script en Python que ejecuta el experimento y mide los tiempos.

- Se ejecutan con los siguientes niveles de recursos:
  - 25%: 1 de 4 procesadores, 5GB de memoria RAM.
  - 50%: 2 de 4 procesadores, 7GB de memoria RAM.
  - 75%: 3 de 4 procesadores, 12GB de memoria RAM.
- Las pruebas se realizaron con la mayor exactitud posible, es decir con la menor cantidad de procesos corriendo en la máquina de prueba.

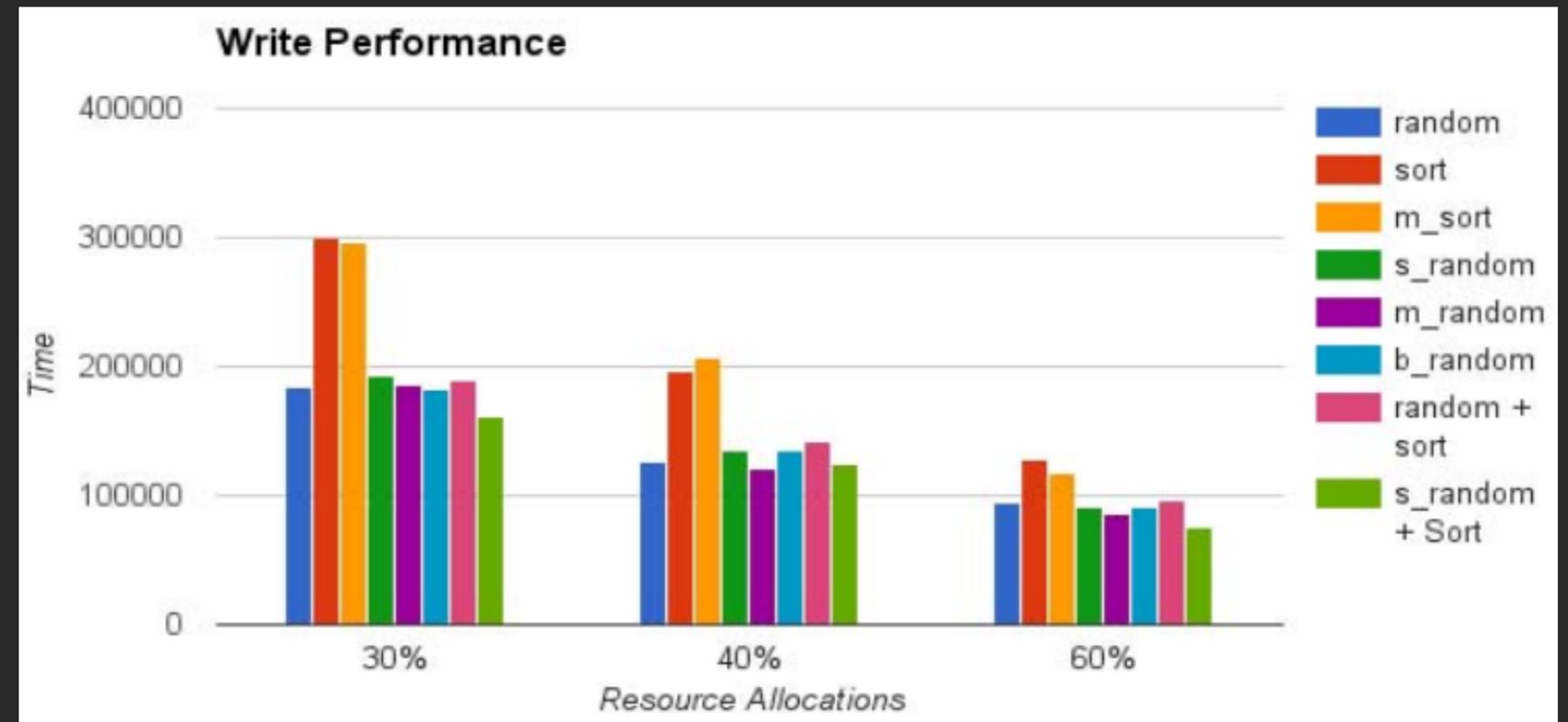
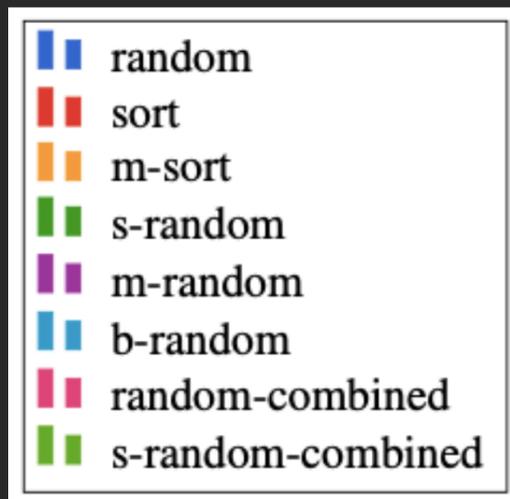
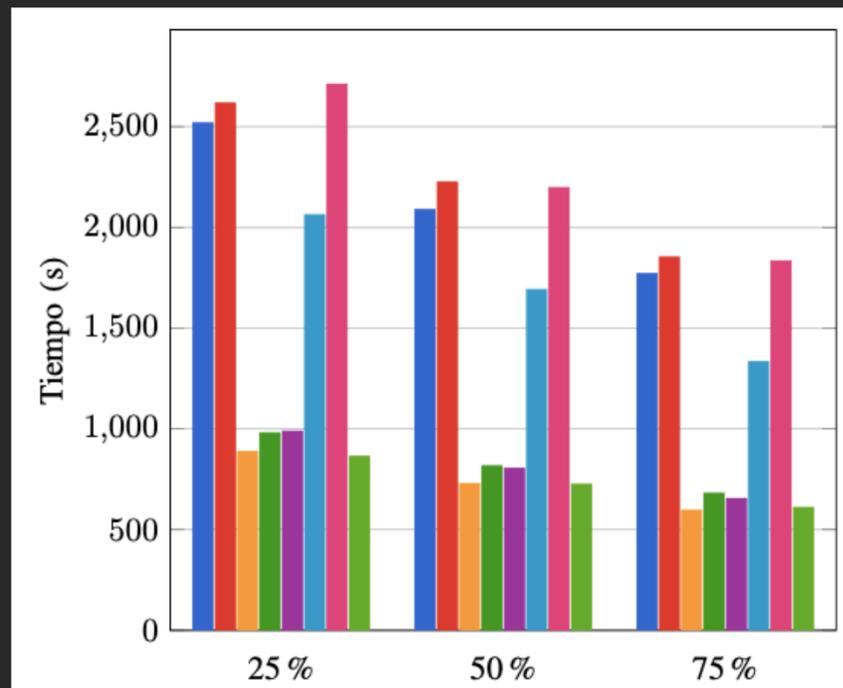
# Resultados de los experimentos



# Conclusiones de los experimentos de escritura

- Ninguna de las claves, ya sea ordenada o aleatoria ofrecen un rendimiento destacado.
- En el caso de claves ordenadas, su rendimiento podría degradarse ya que se puede llegar a sobrecargar a un fragmento, debido a que inserta en el mismo fragmento reiteradamente.
- Estos tiempos para las tres claves fueron bastante similares, aunque un poco mejor para las **claves aleatorias**, en comparación con el trabajo replicado en el que la clave aleatoria tiene un rendimiento significativamente mayor.
- El rendimiento disminuye acorde con sus limitaciones, entonces se puede suponer que en **cualquier nivel de recursos el patrón de rendimiento que afecta a la clave de fragmentación no cambiará.**

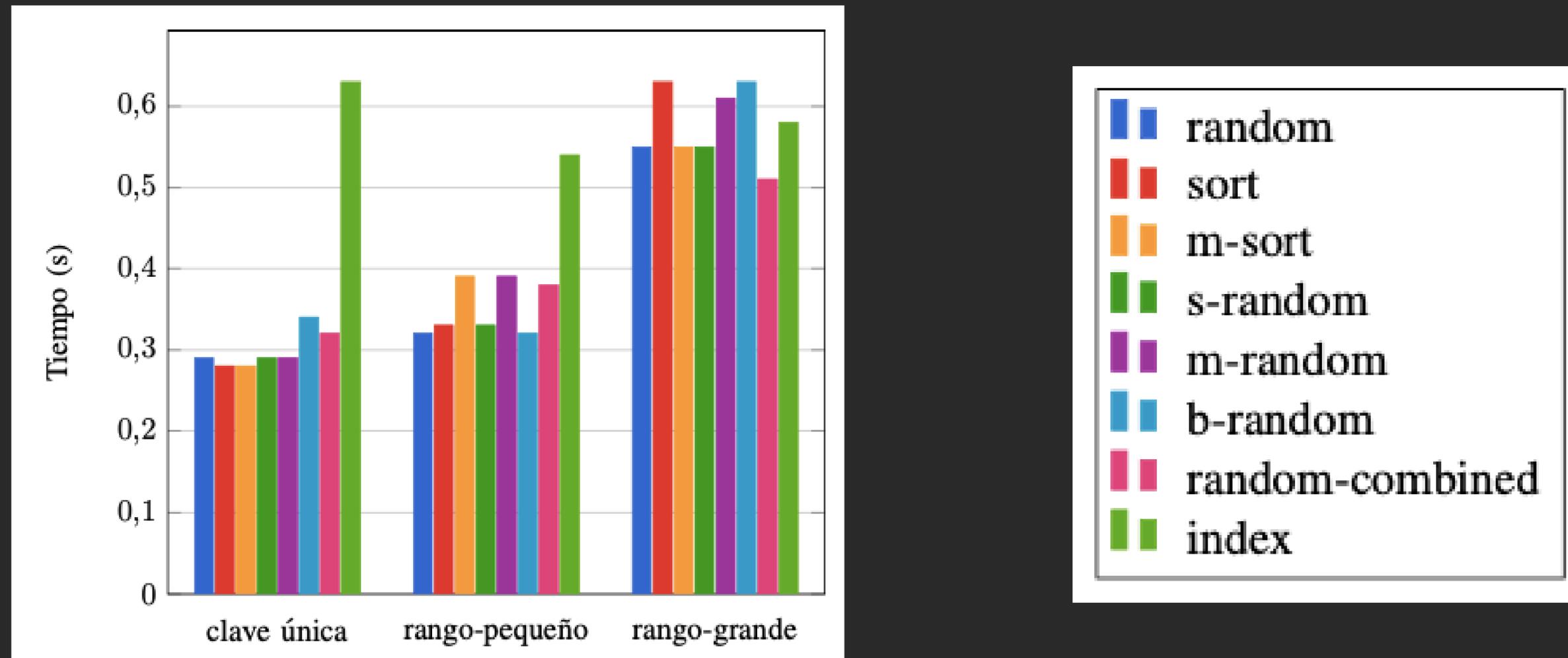
# Comparación de los experimentos de escritura



# Experimentos de lectura

- Realizadas sobre los datos ya insertados por las pruebas de escritura.
- El objetivo de los experimentos de lectura es obtener datos de los distintos fragmentos.
- Para ello se diseñan los experimentos de la siguiente manera:
  - **Consulta de clave única:** solicita un único valor de la clave de fragmentación.
  - **Consulta de clave de rango pequeño:** solicita un pequeño rango de documentos.
  - **Consulta de clave de rango grande:** solicita un gran rango de documentos.
  - **Consulta de clave indexada:** solicita documentos indexados por otro atributo que no es la clave de fragmentación

# Resultados de las pruebas de lectura



# Conclusiones de las pruebas de lectura

- La ejecución de una consulta de **rango más amplio** obtiene un **peor rendimiento** de lectura que una de **rango más pequeño**.
- Las consultas **clave indexada tienen un rendimiento deficiente** en todo tipo de consultas ya que deben acceder a varios fragmentos para responder una consulta, por esto mismo se degrada el rendimiento del sistema.
- Por supuesto, las consultas de clave única o el rango pequeño de búsqueda, tienen mejores rendimiento que las consultas de clave indexada, o las consultas de clave de rango amplio, ya que solicitan una menor cantidad de datos.

# Conclusiones finales

- En comparación con los resultados del trabajo a replicar, se puede concluir que son similares en cuanto a los experimentos de lectura y escritura (las diferencias entre tiempos de ejecución pueden deberse a que los equipos utilizados son distintos, también el hecho de tener distintos conjuntos de datos, etc.).
- En ambos trabajos, se puede concluir que una clave de fragmentación con aleatoriedad y buena localidad puede ofrecer un buen rendimiento tanto de escritura, como de lectura.
- Además, en caso de que una clave de fragmentación tenga valores casi ordenados, la combinación de un pequeño rango de valores aleatorios podría proporcionar un rendimiento aceptable tanto para lectura como para escritura.

# Recursos y trabajos futuros

Queda pendiente el T-Test (prueba t de student) para los experimentos realizados. El T-Test es un tipo de estadística deductiva que se utiliza para determinar si hay una diferencia significativa entre las medias de dos grupos, en este caso sería entre dos experimentos de escritura o lectura.

**¿Preguntas?**