Estudio de rendimiento en MongoDB sobre arquitecturas centralizadas y distribuidas

Autor: Pablo Soldi
Director: Prof. Fernando G. Tinetti
Asesor Profesional: Lic. Franco Agustín Terruzzi
Licenciatura en Sistemas
Facultad de Informática - UNLP

Elección de MongoDB

- Base de datos de tipo NoSQL más utilizada en la actualidad
  - 5° manejador de base de datos más utilizado a nivel global. Encontrándose por debajo de las cuatro opciones SQL más utilizadas: Oracle, MySQL, SQL Server y PostgreSQL
- Colaborativa de código abierto
- Forma de escribir consultas
  - Capitaliza experiencia del lenguaje SQL logrando ciertas semejanzas en la escritura de consultas
- Fácil administración de estructuras complejas
  - Actualización de réplicas y balanceo de cargas en estructuras distribuidas

Objetivos

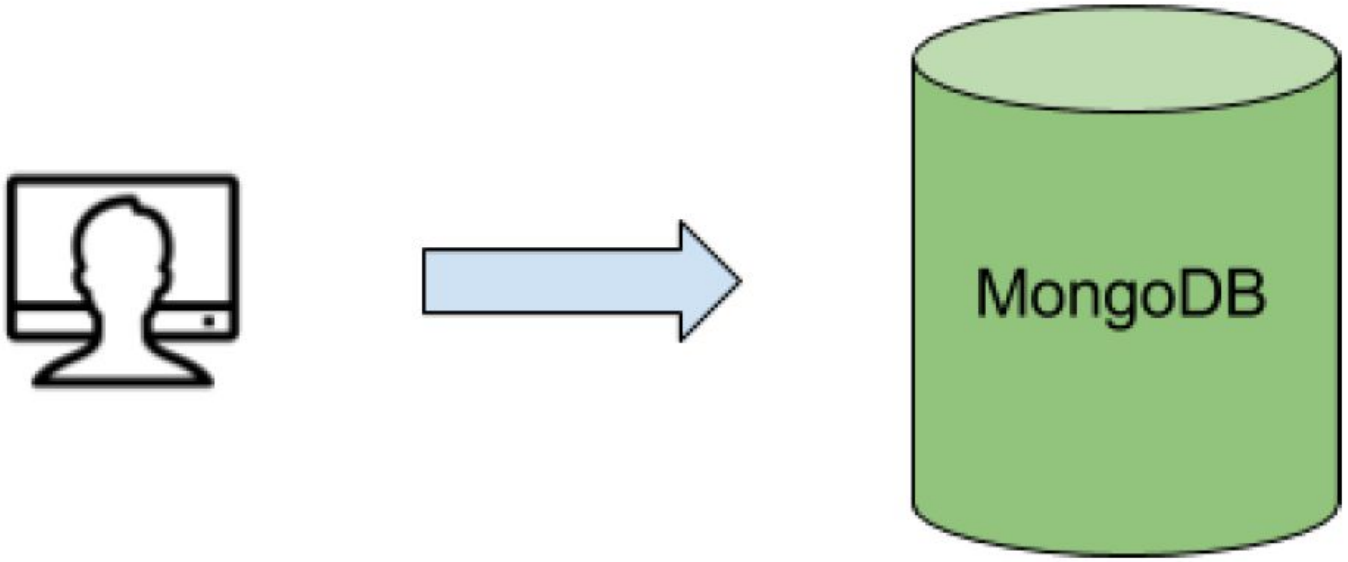
- Conocer y analizar las características de MongoDB para
  - Replicación
  - Distribución
- Rendimiento en diferentes escenarios de almacenamiento de datos
  - No replicados
  - Replicados
  - Distribuidos con replicación

Sucesión de experimentos, validaciones y análisis de resultados

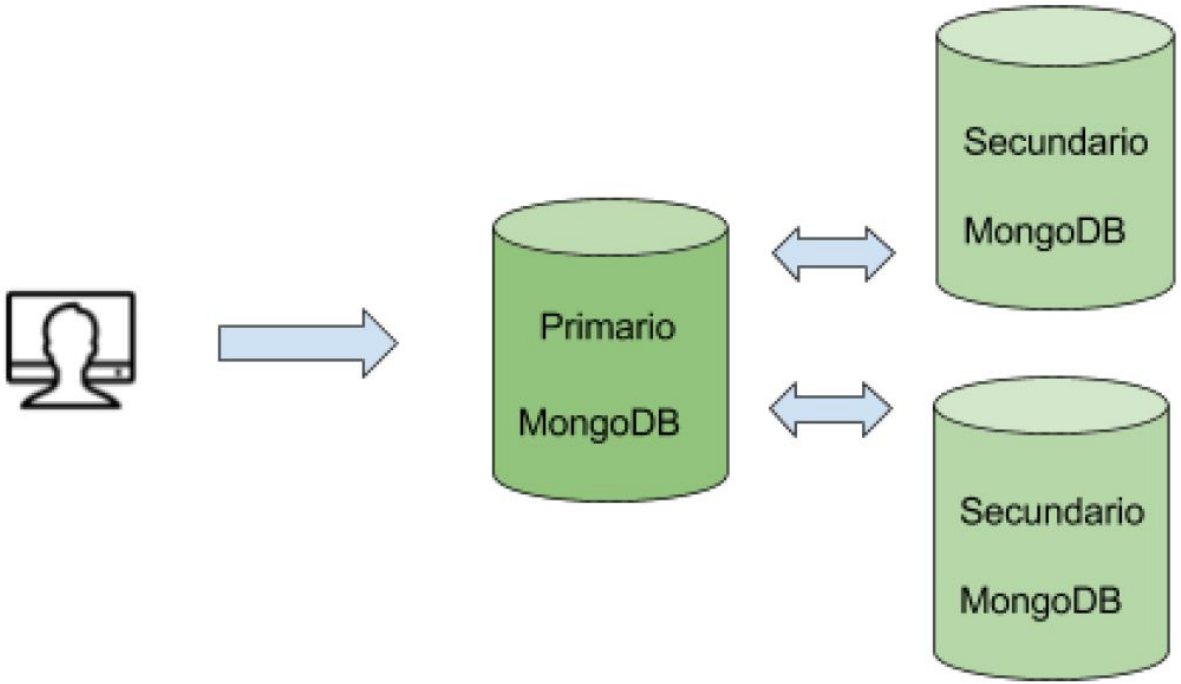
1. Configurar en el entorno de MongoDB distintas estructuras posibles de una base de datos

2. Evaluar distintas métricas sobre el rendimiento
   a. Tiempo de ejecución en cuanto a la inserción
   b. Consistencia de los datos guardados
   c. Tiempo de respuesta para recuperar la información

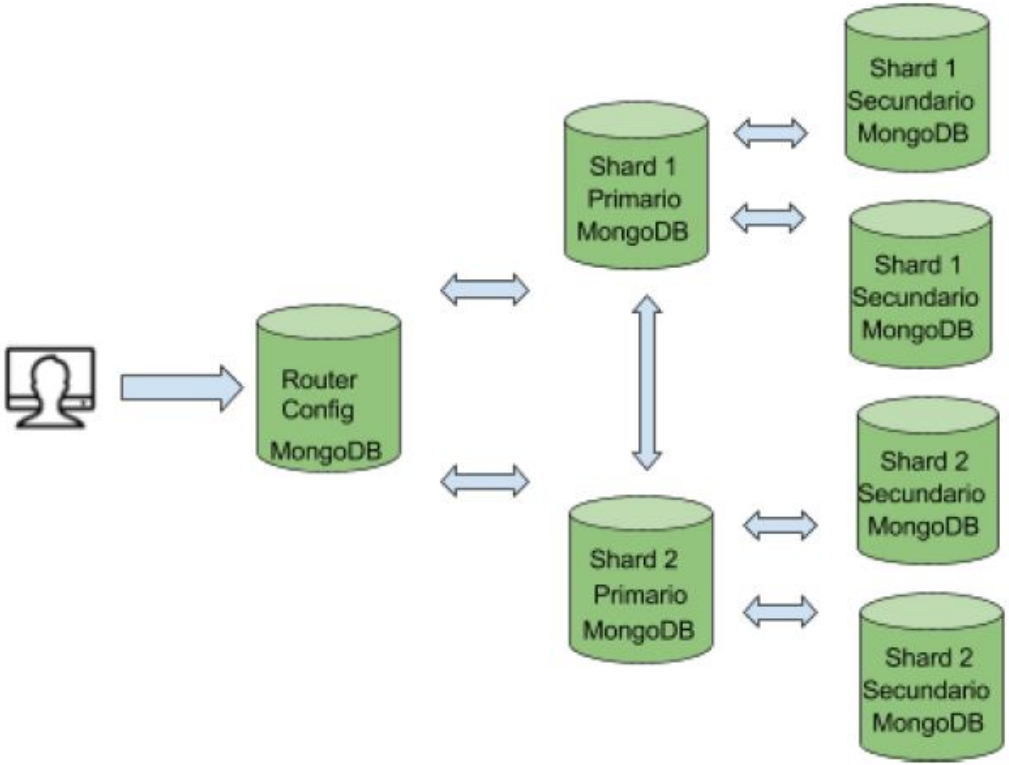
Esquema centralizado



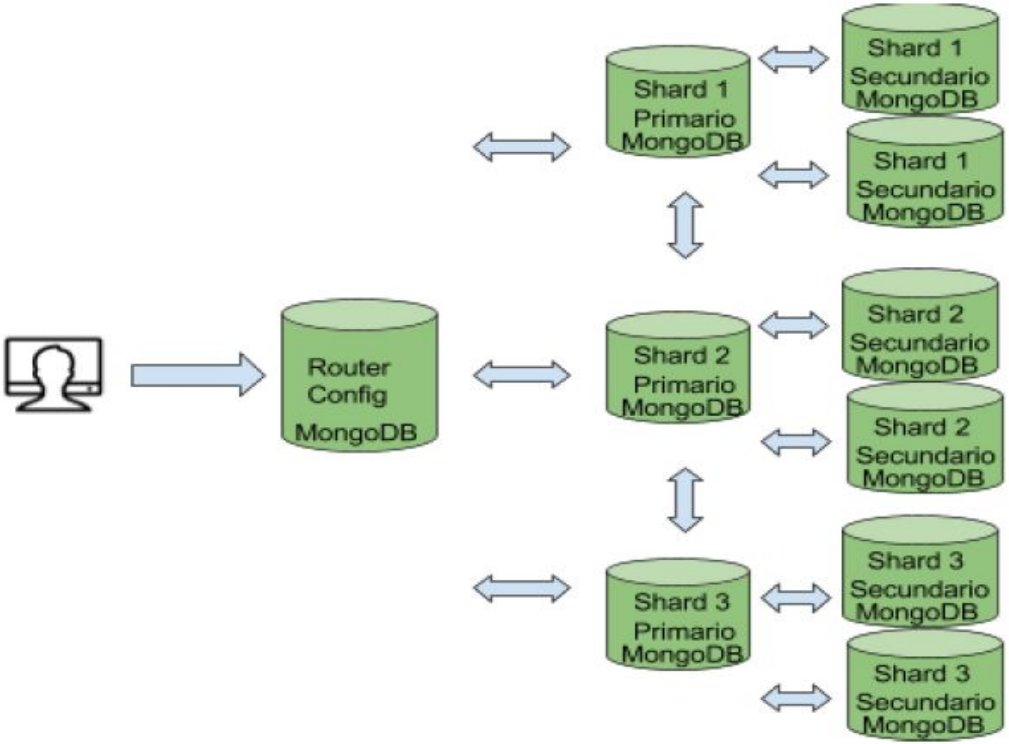
Esquema centralizado con réplicas



Esquema distribuido sobre 2 shards con réplicas



Esquema distribuido sobre 3 shards con réplicas



Implementación - Desarrollo de funcionalidades

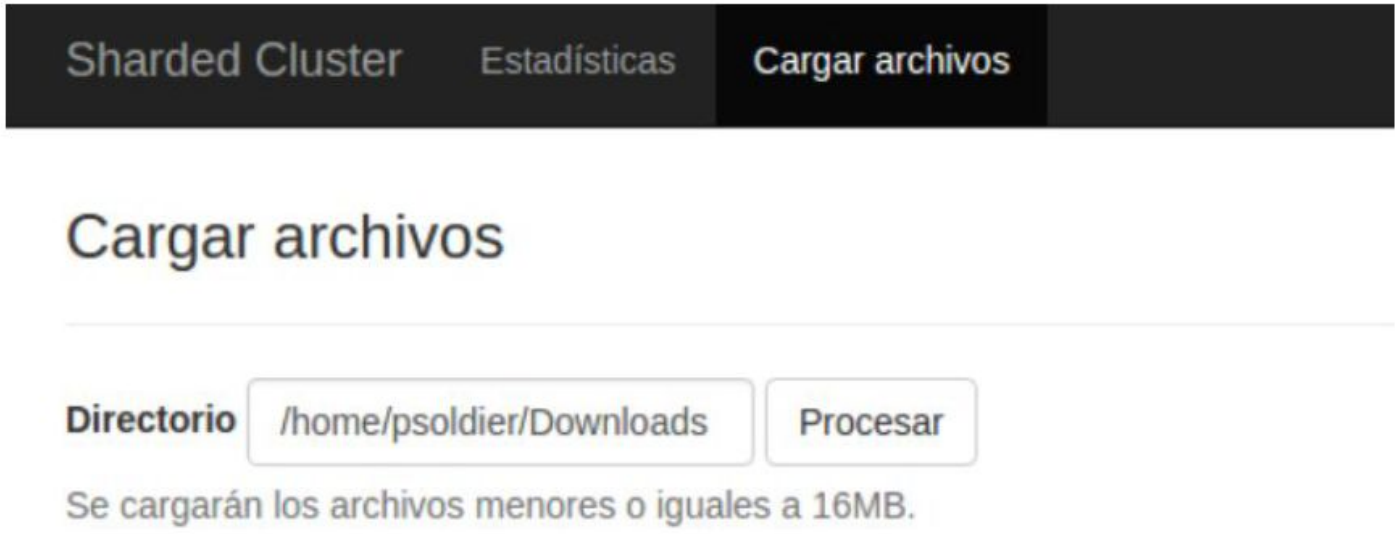
- Inserción de archivos
- Validación de consistencia
- Consultas de lectura de orden lineal
  - Todos los archivos de un tipo particular
  - Archivo específico por nombre
  - Todos los archivos mayores a N bytes
  - Archivos que pertenezcan a un shard determinado (estructuras distribuidas)

Implementación - Desarrollo de funcionalidades

- Registrar en estructura alterna
  - Tiempo de ejecución de la consulta
  - Tamaño de la base de datos al momento de ejecutar la consulta
- Periodicidad en la ejecución de consultas
  - 1GB, 2.5GB, 5GB, 10GB, 20GB

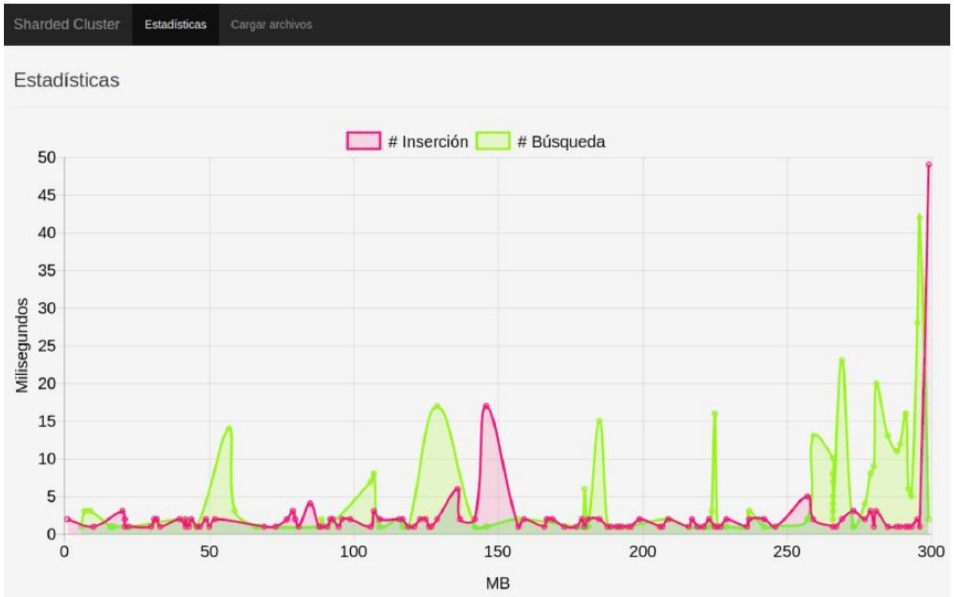
Implementación - Desarrollo de funcionalidades

- Unificar funcionalidades en una única interfaz



Implementación - Desarrollo de funcionalidades

- Unificar funcionalidades en una única interfaz



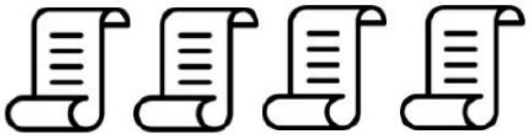
Implementación - Desarrollo de funcionalidades

- Recopilación de datos para los estudios
  - Se necesitaba una colección de archivos de muchos tipos y tamaños diferentes < 16 MB (60GB)
  - Wget + Wikipedia

Implementación - Desarrollo de funcionalidades

Lubuntu

1. Configuración local de las distintas estructuras y posterior documentación de instructivo. Lubuntu + MongoDB en VirtualBox

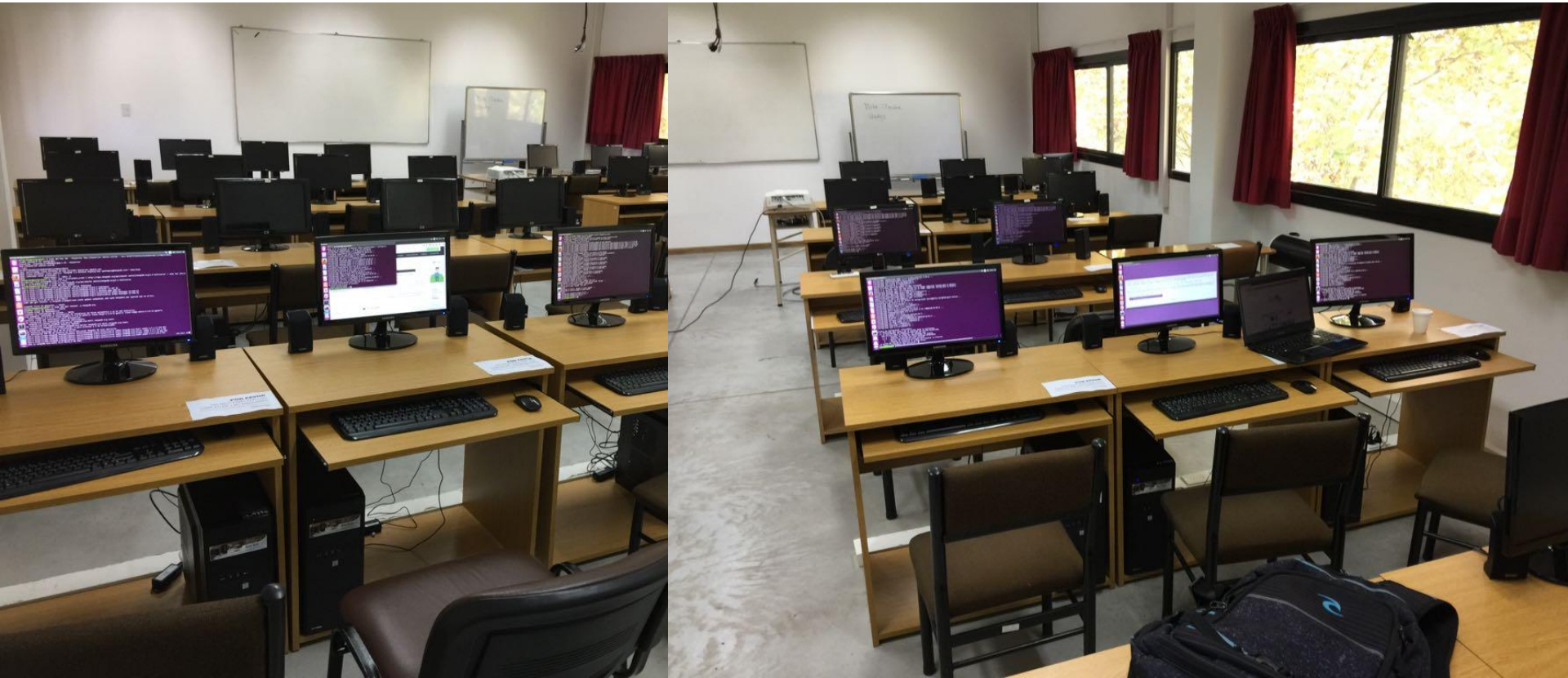


2. Desarrollo de procesos automáticos de las distintas búsquedas, escrituras, validación de consistencia y registro de tiempos de ejecución.

3. Unificación de las funcionalidades desarrolladas bajo una misma interfaz web desarrollada en ruby

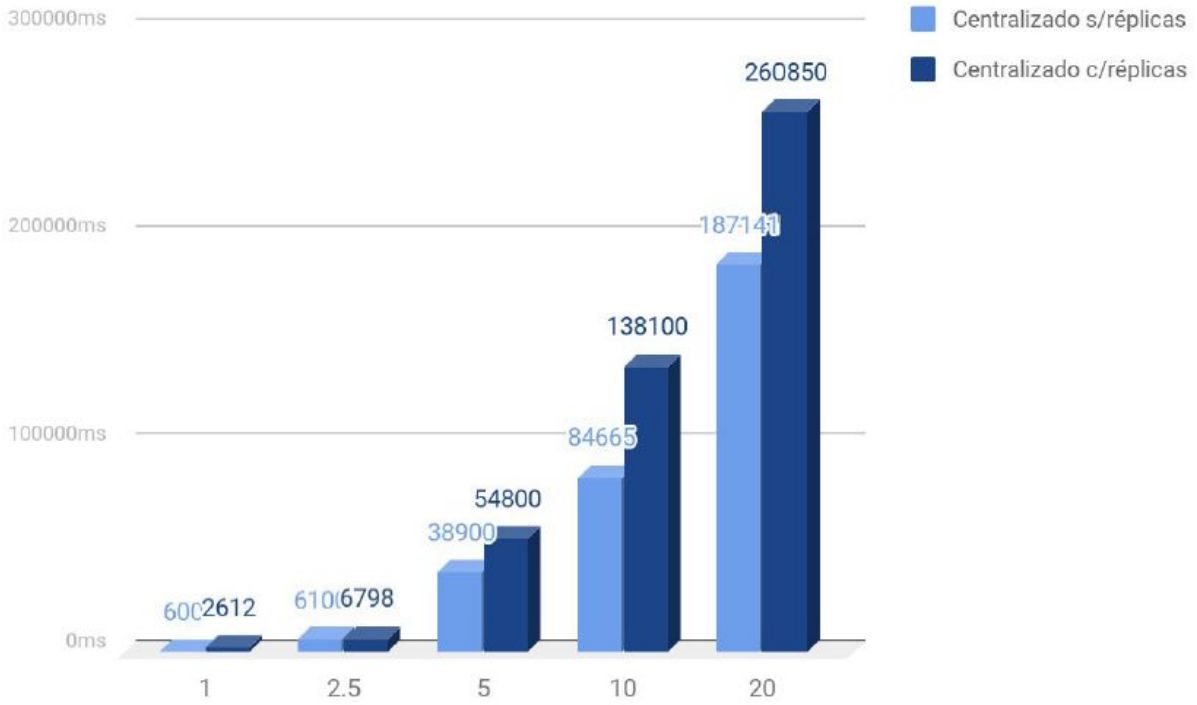
4. Descarga de archivos sobre un disco externo para poder realizar los estudios sobre las diferentes estructuras

Implementación - Desarrollo de funcionalidades



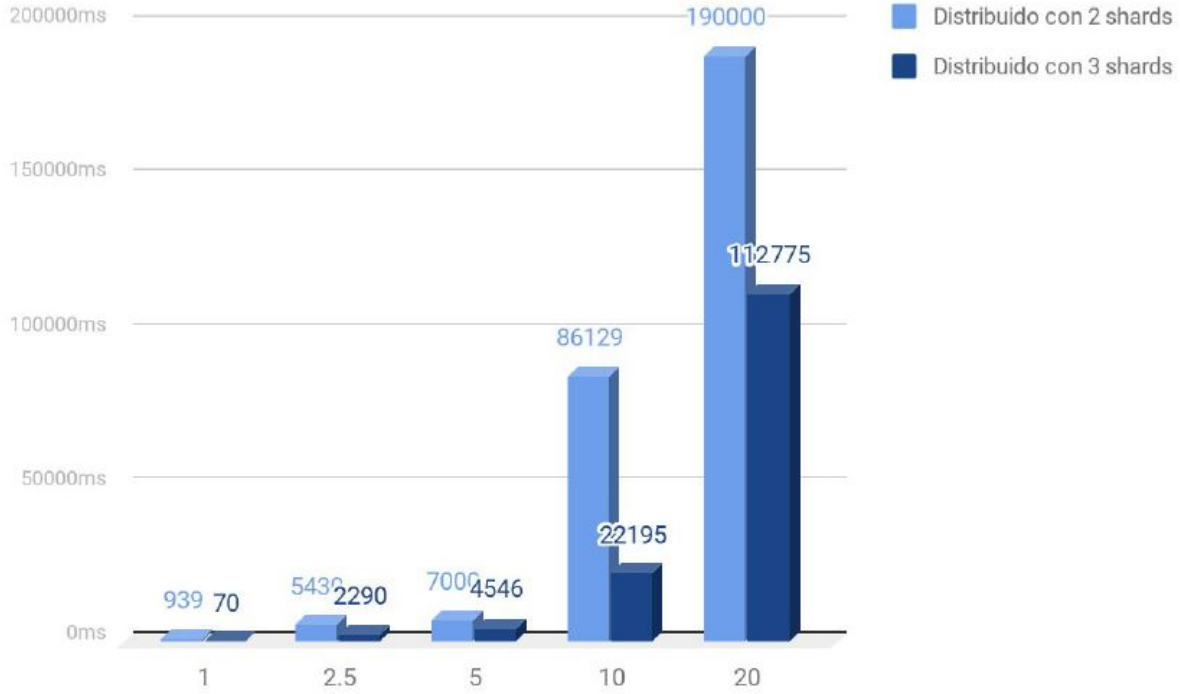
Resultados

Comparación entre estructuras centralizadas



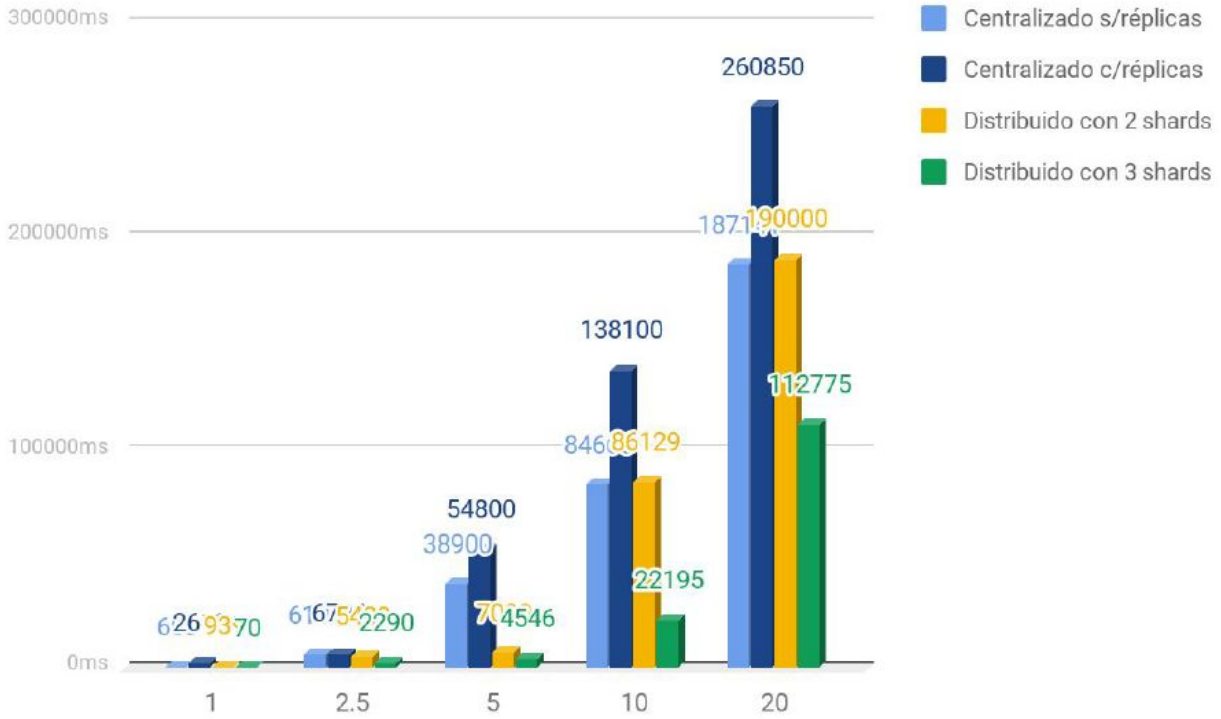
Resultados

Comparación entre estructuras distribuidas



Resultados

Comparación entre todas las estructuras analizadas



Conclusiones

- Arquitecturas distribuidas mejoran los tiempos de respuesta por sobre las centralizadas.
- Comparación entre estructuras distribuidas con 2 y 3 shards permite mejorar prácticamente un 50% los tiempos de respuesta.
- Consistencia en escritura y lectura manejando gran cantidad de datos.
- Curva de aprendizaje costosa para su utilización.

Trabajos futuros

- Comparación entre estructuras distribuidas
  - Hasta qué punto mejora los tiempos de respuesta agregar shards?
- File Server
- Aplicar MongoDB para versionado de código
- Virtualización y MongoDB: Docker, Vagrant
- Comparación con otras opciones NoSQL

¿Preguntas?

Muchas gracias